

INVESTIGATION OF QUALITY INDICES OF IN-PARALLEL PLATFORM  
MANIPULATORS AND DEVELOPMENT OF WEB BASED ANALYSIS TOOL

By

JAEHOON LEE

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2000

Copyright 2000

by

Jaehoon Lee

To my lovely wife Inhee .

## ACKNOWLEDGMENTS

The author thanks God who is taking care of his life and his wife, Inhee, for her encouragement and support and love throughout the process of this research. He wishes to express his deep appreciation to Dr. Joseph Duffy for his invaluable guidance, encouragement and patience during the time of this research and would like to acknowledge the important roles played by Dr. Kenneth Hunt in the development of the contents of this thesis.

In addition, he thanks Dr. Neil White for help with the author's understanding of the underlying mathematical concepts and thanks Dr. Carl D. Crane, Dr. John Ziegert and Dr. Ralph Selfridge for their instructions and supervision in several courses and works. He also thanks all his friends, specially Arfath Pasha, for their help and support.

Finally, he thanks his family who is always praying for his well being.

## TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENTS .....	iv
LIST OF FIGURES .....	vii
ABSTRACT .....	x
CHAPTERS	
1 INTRODUCTION .....	1
2 DETERMINATION OF $ \det J _m$ FOR A 3-3 OCTAHEDRAL MANIPULATOR.....	6
3 A GEOMETRICAL INTERPRETATION OF $\det J$ .....	14
4 FURTHER DEPARTURES FROM THE CENTRAL CONFIGURATION .....	18
5 DETERMINATION OF $ \det J _m$ FOR A 3-6 PLATFORM MANIPULATOR .....	24
6 DETERMINATION OF $ \det J _m$ FOR A 6-6 PLATFORM MANIPULATOR .....	29
7 COMPATABILITY OF THE RESULTS OF THE 3-3, 3-6, 6-6 PLATFORMS .....	35
8 OTHER ALTERNATIVE JOINT SEPARATIONS .....	38
<i>Joint Separation toward the Center of the Platform</i> .....	38
<i>Joint Separation along One Side of the Triangle</i> .....	48
9 DEVELOPMENT OF WEB BASED INTERACTIVE GRAPHICAL DESIGN AND ANALYSIS TOOL .....	57
<i>Specifications</i> .....	58
Hardware.....	58
Software .....	58
<i>Programming Languages and tools</i> .....	59
VRML.....	59
Java and JavaScript.....	59
VBScript .....	60

Active Server Page (ASP).....	60
Remote Scripting .....	60
ODBC (Open DataBase Connectivity) .....	61
<i>System Design</i> .....	61
<i>Procedures</i> .....	63
Setup Client Side Components (Integrating Java and VRML) .....	63
Setup Server Side Components.....	64
<i>Results</i> .....	66
10 CONCLUSION .....	69
APPENDIX PROGRAM LISTS .....	71
LIST OF REFERENCES .....	125
BIOGRAPHICAL SKETCH.....	127

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1 Equilateral triangular platform in its central location, planar motion .....	5
2 Octahedral 3-3 manipulator, plan view .....	6
3 Octahedron in the central configuration (highest quality index), plan view .....	12
4 How quality index varies with distance between base and platform .....	13
5 Coplanar translation of platform from central location : contours of quality index .....	19
6 Rotation of platform about Z-axis .....	21
7 A singularity configuration (Z axis rotation) .....	22
8 Rotation of platform about X-axis .....	22
9 Rotation of platform about Y-axis .....	23
10 A singularity configuration (X axis rotation) .....	23
11 Spatial 3-6 platform device (plan view) .....	24
12 Spatial 3-6 platform in optimal configuration .....	28
13 Spatial 6-6 platform device (plan view) .....	29
14 Plot of $f(\alpha, \beta) = 3\alpha\beta - 2\alpha - 2\beta + 1 = 0$ and configurations for $\det J = 0$ .....	32
15 Spatial 6-6 platform in optimal configuration .....	34
16 Compatibility between the 3-3 octahedral platform and the 3-6 platform .....	37
17 Reduction of the size of the moving platform when $\alpha = 1/2$ .....	37
18 Platform with the joints separated toward the center .....	39

19	$\det J_m$ vs $\rho$ (platform with joints separated toward the center).....	42
20	Z-axis rotation vs $\rho$ (platform with joints separated toward the center).....	43
21	$h$ vs $\rho$ (platform with joints separated toward the center) .....	43
22	Optimal configuration of the platform with the joints separated toward the center ....	44
23	Variation of quality index on x-y plane : contours of quality index (platform with joints separated toward the center).....	45
24	Variation of quality index along Z-axis (platform with joints separated toward the center) .....	46
25	Variation of quality index when platform rotates about X-axis (platform with joints separated toward the center) .....	46
26	Variation of quality index when platform rotates about Y-axis (platform with joints separated toward the center) .....	47
27	Variation of quality index when platform rotates about Z-axis (platform with joints separated toward the center) .....	47
28	Platform with joints separated along one edge of the triangles .....	49
29	$\det J_m$ vs $\rho$ (platform with joint separated along one edge of the triangle).....	51
30	Z-axis rotation vs $\rho$ (platform with joint separated along one edge of the triangle) ...	52
31	$h$ vs $\rho$ (platform with joint separated along one edge of the triangle) .....	52
32	Optimal configuration of platform with joints separated along one edge of the triangle .....	53
33	Variation of quality index on x-y plane : contours of quality index (platform with joint separated along the edges of the triangle).....	54
34	Variation of quality index along Z-axis (platform with joint separated along one edge of the triangle).....	55
35	Variation of quality index when platform rotates about X-axis (platform with joint separated along one edge of the triangle).....	55
36	Variation of quality index when platform rotates about Y-axis (platform with joint separated along one edge of the triangle).....	56



37	Variation of quality index when platform rotates about Z-axis (platform with joint separated along one edge of the triangle).....	56
38	System Diagram.....	62
39	Database table information .....	65
40	Web based interactive analysis tool for platform manipulators.....	67
41	Platform configuration dialog .....	68
42	Dialog displaying Jacobian Matrix .....	68

Abstract of Dissertation Presented to the Graduate School  
of the University of Florida in Partial Fulfillment of the  
Requirements for the Degree of Doctor of Philosophy

INVESTIGATION OF QUALITY INDICES OF IN-PARALLEL PLATFORM  
MANIPULATORS AND DEVELOPMENT OF WEB BASED ANALYSIS TOOL

By

Jaehoon Lee

August 2000

Chairman: Dr. Joseph Duffy

Major Department: Mechanical Engineering

A platform manipulator is a fully in-parallel device. It has a linear actuator on each of its six legs. The legs connect a platform to a base in a certain pattern that allows each leg to sustain a load from the platform. Design of leg configurations is not an easy task. Some configurations may lead a manipulator into an uncontrollable state known as a singularity.

The proposed 'quality index' takes a maximum value of 1 at an optimal configuration that is shown to correspond to the maximum value of the determinant of the six-by-six Jacobian matrix of the manipulator. When the manipulator is actuated so that the platform departs from this optimal configuration, the determinant always diminishes, and, as is well known, it becomes zero when a special configuration is reached (the platform then gaining one or more uncontrollable freedoms). The quality index  $\lambda$ ,  $0 \leq \lambda \leq 1$ , is a constructive measure of acceptable design proportions.

The octahedral manipulator, which has equilateral triangles in both the platform and the base, turns out to be the most geometrically stable design (strongest design) of a manipulator. However, the double-spherical joints—and there are six of them—are the source of the critical practical difficulties. Kinematic substitutions can circumvent this problem, but most often there is no reasonable alternative to accepting separations of some or all of the double-ball joints. Several patterns of joint separation have been suggested in the paper and each of them is investigated to obtain optimal configuration.

## CHAPTER 1 INTRODUCTION

The Jacobian Matrix (at a configuration of a fully in-parallel robotic device with one linear actuator per leg) serves two distinct purposes that, in the terminology of screw theory, are mutually reciprocal.

In its ordinary form as a six-by-six matrix the columns of which are the coordinates of the actuator screws (ordinarily the normalized leg-lines), it is possible to obtain from six actuated force inputs the wrench that the end-effector platform delivers. In its transposed form the Jacobian can give the relative linear speeds required at each linear actuator that correspond to a given twist required to be executed by the platform. The first of these gives the (instantaneous) solution to a problem of static equilibrium; the second, the solution of first-order kinematic compatibility. When the Jacobian is singular (i.e., when its determinant is zero) the actuators (i) cannot equilibrate a general wrench applied to the platform and (ii) cannot, on their own, prevent a transitory uncontrollable movement of the platform. This latter phenomenon is associated with the platform's gain of one or more freedoms when all actuators are locked.

When the Jacobian is singular the corresponding manipulator-configuration is said to be *special*. There are other manifestations of special configurations, for instance, when an actuator reaches its end of travel (the platform then ordinarily loses a freedom.) Such situations need careful attention when the input variable is, say a rotation at the same joint that serves to extend or shorten a kinematically equivalent linear-

actuator substitution in a ball-jointed leg; with suitable modifications (and some care) such structures could come within the range discussed here.

It is mentioned in passing the work of Cox (1981) and Duffy (1996), both of whom cover special configurations of planar-motion manipulators. Merlet and Gosselin (1991), amongst others, examine the special configurations of a new architecture of a six-degree-of-freedom parallel manipulator.

Hunt and McAree (1998) go into considerable detail for the general octahedral manipulator, its special configurations being described in the context of other geometrical properties. The condensed form of the Jacobian matrix in McAree and Daniel (1996) may, we believe, be used as a basis from which to develop the argument that follows.

From the practical standpoint and in the absence of additional control means, the manipulator must not approach “too closely to” a special configuration and the question “how close is too close?” is often hard to answer.

Here the problem is tackled from the other end. Both the proportions and the configuration of that octahedral manipulator that gives the greatest absolute value of the determinant of the Jacobian  $J$ , namely  $|\det J|_m$ , is determined. Because at any other configuration  $|\det J|$  takes a lower value, a dimensionless quality index defined by

$$\lambda = \frac{|\det J|}{|\det J|_m} \quad (1)$$

at every reachable configuration,  $0 \leq \lambda \leq 1$ .

Some may wonder why the equation (1) is adopted as an index of quality rather than use other well-established methods (found in books on theory of matrices and linear

algebra) that lead, via norms, or diagonalization and singular value decomposition, and so forth, to properties that relate to ‘conditioning’.

Such methods are nearly always based on the presumption that a column-vector, say, of a six-by-six matrix can be treated as a vector in  $\mathcal{R}^6$ . However, the six elements in the column of a typical robot Jacobian are the normalized coordinates of a screw (almost always of zero pitch, i.e., a line); in a metrical coordinate frame three of them are dimensionless and three have dimension [length], such a length being the measure of the moment about a reference point of a unit force.

The column is made up of two distinct vectors, each of them in  $\mathcal{R}^3$ , except in two circumstances, in both of which the Jacobian is reduced to a three-by-three, pure translational velocity in any direction, or pure angular velocity about any axis *through the origin*. In general there is no way of unifying the dimensions of the coordinates of a normalized line. For the legs of the octahedral manipulator it is not possible to remove all the length dimensions from their coordinates because, even the adoption of some artificial length unit can fail due to the fact that a moment can never be converted to a pure force. Moreover, any index of quality derived from such textbook techniques is likely to vary according to the coordinate frame in which the Jacobian is formulated.

I do not here examine the circumstances in which such an approach *may* produce meaningful results; rather we remain on what we know is safe ground, recognizing first, that the determinant of a (square) Jacobian of line coordinates (or even of the more general screw coordinates) is dependent solely on the configuration in  $\mathcal{R}^3$  of the actuated axes and not at all on the coordinate frame in which the line coordinates are determined;

and second, that because equation (1) is a dimensionless ratio, our quality index is always independent of the choice of units of length measurement.

For the planar-motion manipulator of Figure 1 when the platform triangle is general, its sides having lengths  $l_1$ ,  $l_2$ , and  $l_3$ , Lee, Duffy and Keler (1996) found  $|\det J|_m$  to be equal to  $\frac{1}{2}(l_1 + l_2 + l_3)$ . When the platform is equilateral, as drawn,  $|\det J|_m = \frac{3}{2}l$ .

With suitable fixed base-point  $E_A$ ,  $E_B$  and  $E_C$  placed anywhere on the sides of the larger triangle (shown in dotted outline), as triangle  $ABC$  is guided away from its central location the value of  $\lambda$  (see equation (1)) always diminishes.

In what follows, I make a comparable study of the (spatial) octahedral manipulator. Having determined the proportion and the configuration that give  $\lambda = 1$ , then I examine the extent to which  $\lambda$  degrades as the platform displaces.

Further, I extend the research to a 3-6 platform device which has three double joints on a moving platform and six single joints on the base and 6-6 platform device which has six single joints on both moving and base platform.

A quality index has two clear meanings so far. When  $\lambda=0$ , a platform is in singular condition and when  $\lambda=1$ , it is in its optimal configuration to sustain loads. However, when  $\lambda$  is neither zero nor one, it is hard to say exactly how much one configuration is better than another. One can not say that a configuration with  $\lambda=0.8$  is twice as good as a configuration with  $\lambda=0.4$  without further analyses on the matter. But, a quality index helps to design platforms by setting dimensions that give best quality index value i.e.  $\lambda=1$ . Also, it gives an idea of certain designs that must be prevented as they would lead to zero or low quality indexes.

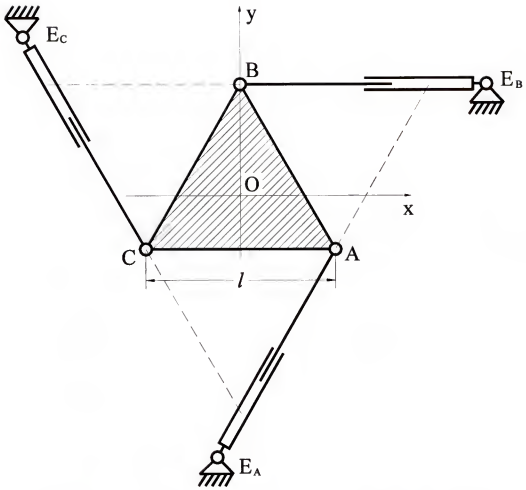


Figure 1 : Equilateral triangular platform in its central location, planar motion



CHAPTER 2  
DETERMINATION OF  $|\det J|_m$  FOR A 3-3 OCTAHEDRAL MANIPULATOR

Figure 2 illustrates the plan view of a 3-3 device with an equilateral triangular base of side  $b$ , and an equilateral triangular moving platform of side  $a$ . The moving platform is parallel to the base and a distance  $h$  from it. The six legs  $E_A A$ ,  $E_B A$ ,  $E_B B$ ,  $E_C B$ ,  $E_C C$ , and  $E_A C$  have ball-joints at their ends and linear actuators to vary their lengths.

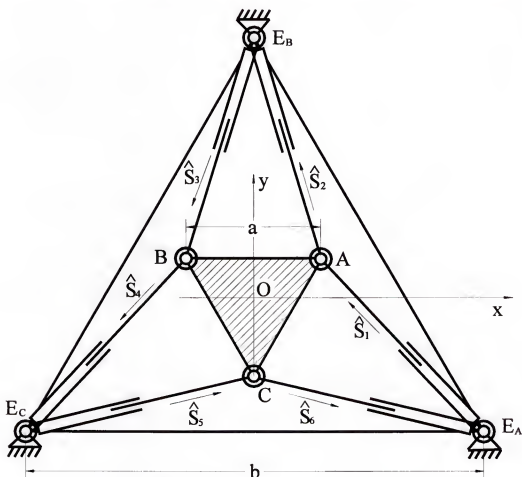


Figure 2 : Octahedral 3-3 manipulator, plan view

It is required to determine the value of  $h$  which yields a maximum absolute value for  $|\det J|$ , the determinant of the line coordinates of the six legs, that is an expression for  $|\det J|_m$ .

The  $(x, y, z)$  coordinates of the points  $A, B, C, E_A, E_B$  and  $E_C$  are determined with the origin  $O$  of a fixed coordinate frame placed at the center of the triangle  $ABC$ , and then

$$\begin{aligned} A \begin{pmatrix} \frac{a}{2} & \frac{a}{2\sqrt{3}} & 0 \end{pmatrix}, \quad B \begin{pmatrix} -\frac{a}{2} & \frac{a}{2\sqrt{3}} & 0 \end{pmatrix}, \quad C \begin{pmatrix} 0 & -\frac{a}{\sqrt{3}} & 0 \end{pmatrix}, \\ E_A \begin{pmatrix} \frac{b}{2} & -\frac{b}{2\sqrt{3}} & -h \end{pmatrix}, \quad E_B \begin{pmatrix} 0 & \frac{b}{\sqrt{3}} & -h \end{pmatrix}, \quad E_C \begin{pmatrix} -\frac{b}{2} & -\frac{b}{2\sqrt{3}} & -h \end{pmatrix}. \end{aligned} \quad (2)$$

The coordinates of a line joining two finite points with coordinates  $(x_1, y_1, z_1)$  and  $(x_2, y_2, z_2)$  are given by the second-order determinants of the array

$$\begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \end{bmatrix}. \quad (3)$$

The direction ratios of the line are

$$L = \begin{vmatrix} 1 & x_1 \\ 1 & x_2 \end{vmatrix}, \quad M = \begin{vmatrix} 1 & y_1 \\ 1 & y_2 \end{vmatrix}, \quad N = \begin{vmatrix} 1 & z_1 \\ 1 & z_2 \end{vmatrix},$$

and the moment of the line segment about the three coordinate axes are

$$P = \begin{vmatrix} y_1 & z_1 \\ y_2 & z_2 \end{vmatrix}, \quad Q = \begin{vmatrix} z_1 & x_1 \\ z_2 & x_2 \end{vmatrix}, \quad R = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix}. \quad (4)$$

For example, the coordinates of  $E_A$  are obtained from

$$\begin{bmatrix} 1 & \frac{b}{2} & -\frac{b}{2\sqrt{3}} & -h \\ 1 & \frac{a}{2} & \frac{a}{2\sqrt{3}} & 0 \end{bmatrix}, \quad (5)$$

and therefore,

$$\hat{S}_1 \equiv \left[ \frac{a-b}{2} \quad \frac{a+b}{2\sqrt{3}} \quad h ; \frac{ah}{2\sqrt{3}} \quad -\frac{ah}{2} \quad \frac{ab}{2\sqrt{3}} \right]. \quad (6)$$

The coordinates of  $E_B A$  are, similarly,

$$\hat{S}_2 \equiv \left[ -\frac{a}{2} \quad \frac{2b-a}{2\sqrt{3}} \quad -h ; -\frac{ah}{2\sqrt{3}} \quad \frac{ah}{2} \quad \frac{ab}{2\sqrt{3}} \right]. \quad (7)$$

The remaining coordinates are

$$\hat{S}_3 \equiv \left[ -\frac{a}{2} \quad \frac{a-2b}{2\sqrt{3}} \quad h ; \frac{ah}{2\sqrt{3}} \quad \frac{ah}{2} \quad \frac{ab}{2\sqrt{3}} \right], \quad (8)$$

$$\hat{S}_4 \equiv \left[ \frac{a-b}{2} \quad -\frac{a+b}{2\sqrt{3}} \quad -h ; -\frac{ah}{2\sqrt{3}} \quad -\frac{ah}{2} \quad \frac{ab}{2\sqrt{3}} \right], \quad (9)$$

$$\hat{S}_5 \equiv \left[ \frac{b}{2} \quad \frac{b-2a}{2\sqrt{3}} \quad h ; -\frac{ah}{\sqrt{3}} \quad 0 \quad \frac{ab}{2\sqrt{3}} \right], \quad (10)$$

$$\hat{S}_6 \equiv \left[ \frac{b}{2} \quad \frac{2a-b}{2\sqrt{3}} \quad -h ; \frac{ah}{\sqrt{3}} \quad 0 \quad \frac{ab}{2\sqrt{3}} \right]. \quad (11)$$

Hence, from (6) to (11), the *normalized* determinant of the six lines (now all reduced to unit length) is

$$\det J = \frac{1}{l^6} \begin{vmatrix} \frac{a-b}{2} & -\frac{a}{2} & -\frac{a}{2} & \frac{a-b}{2} & \frac{b}{2} & \frac{b}{2} \\ \frac{a+b}{2\sqrt{3}} & \frac{2b-a}{2\sqrt{3}} & \frac{a-2b}{2\sqrt{3}} & -\frac{a+b}{2\sqrt{3}} & \frac{b-2a}{2\sqrt{3}} & \frac{2a-b}{2\sqrt{3}} \\ h & -h & h & -h & h & -h \\ \frac{ah}{2\sqrt{3}} & -\frac{ah}{2\sqrt{3}} & \frac{ah}{2\sqrt{3}} & -\frac{ah}{2\sqrt{3}} & -\frac{ah}{\sqrt{3}} & \frac{ah}{\sqrt{3}} \\ -\frac{ah}{2} & \frac{ah}{2} & \frac{ah}{2} & -\frac{ah}{2} & 0 & 0 \\ \frac{ab}{2\sqrt{3}} & \frac{ab}{2\sqrt{3}} & \frac{ab}{2\sqrt{3}} & \frac{ab}{2\sqrt{3}} & \frac{ab}{2\sqrt{3}} & \frac{ab}{2\sqrt{3}} \end{vmatrix} \quad (12)$$

Conveniently, here, the normalization divisor is the same for each leg, namely the leg length,  $l = E_A A = E_B A = E_B B = E_C B = E_C C = E_A C$ , and for every leg

$$l = \sqrt{L^2 + M^2 + N^2} = \sqrt{\frac{1}{3}(a^2 - ab + b^2 + 3h^2)}. \quad (13)$$

Expansion of (12) and inclusion of (13) leads to

$$|\det J| = \frac{3\sqrt{3}a^3b^3h^3}{4\left(\frac{a^2 - ab + b^2}{3} + h^2\right)^3}. \quad (14)$$

Dividing above and below by  $h^3$  yields

$$|\det J| = \frac{3\sqrt{3}a^3b^3}{4\left(\frac{a^2 - ab + b^2}{3h} + h\right)^3}. \quad (15)$$

Differentiating the denominator with respect to  $h$  and equating to zero to obtain an optimal value yields

$$h = h_m = \sqrt{\frac{1}{3}(a^2 - ab + b^2)}, \quad (16)$$

which is now substituted in (15) to give the expression for the maximum of  $|\det J|$ ,

namely

$$|\det J|_m = \frac{27a^3b^3}{32(a^2 - ab + b^2)^{\frac{3}{2}}}. \quad (17)$$

Substituting  $b=\gamma a$  into (18) and dividing above and below by  $\gamma$  yields

$$|\det J|_m = \frac{27a^3}{32\left(\frac{1}{\gamma^2} - \frac{1}{\gamma} + 1\right)^{\frac{3}{2}}}. \quad (18)$$

The absolute maximum value of  $|\det J|_m$ , namely  $|\det J|_{am}$ , is obtained by taking the derivative of (18) with respect to  $\gamma$  which yields

$$\frac{1}{\gamma^2}\left(1 - \frac{2}{\gamma}\right) = 0, \quad (19)$$

whence we obtain a further condition, namely

$$\gamma = \frac{b}{a} = 2. \quad (20)$$

This octahedron has, therefore, a maximum quality-index configuration as it is shown in Figure 3, and the distance from the base to the platform is, from (16),  $h=a$ .

Now, from (18)

$$|\det J|_{am} = \frac{3\sqrt{3}}{4}a^3. \quad (21)$$

The volume of the octahedron of Figure 3 is

$$V = \frac{\sqrt{3}}{12} h(a+b)^2 \quad (22)$$

and when  $b=2a$  and  $h=a$ ,

$$V = \frac{3\sqrt{3}}{4} a^3 = |\det J|_{am}. \quad (23)$$

Now the interpretation of  $|\det J|_{am}$  (21) is clear. It is simply the volume of the particular octahedron with moving platform sides  $a$ , base sides  $2a$ , and distance between platform and base  $a$  (Figure 3). This result is comparable to  $|\det J|_{am}$  for the planar-motion equilateral triangular manipulator case for which  $|\det J|_{am} = \frac{3}{2} l$  (see Figure 1).

The quality index  $\lambda = \frac{|\det J|}{|\det J|_m}$  (equation 1) can, from (14) and (17), be expressed

in the form

$$\lambda = \frac{8 \left( \frac{h}{h_m} \right)}{\left\{ 1 + \left( \frac{h}{h_m} \right)^2 \right\}^3} = \frac{8 \left( \frac{h_m}{h} \right)}{\left\{ 1 + \left( \frac{h_m}{h} \right)^2 \right\}^3}, \quad (24)$$

where  $h_m$  is obtained from (16). A plot of  $\lambda$  vs.  $h/h_m$  is shown in Figure 4, and, from (24),

$\lambda = 0$  when  $h = 0$  or  $h = \infty$ .

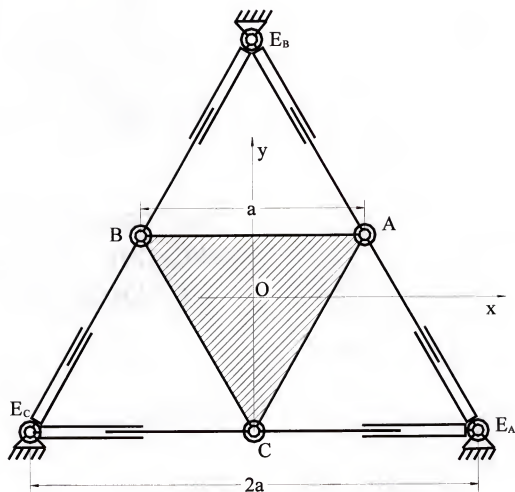


Figure 3 : Octahedron in the central configuration (highest quality index), plan view

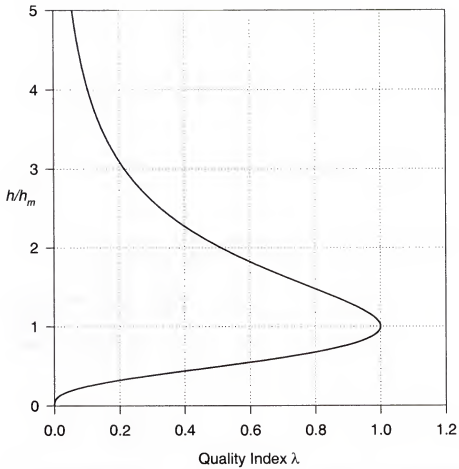


Figure 4 : How quality index varies with distance between base and platform



### CHAPTER 3

#### A GEOMETRICAL INTERPRETATION OF $\det J$

It can be shown using the Grassmann-Cayley algebra (McMillan 1990, White and Whiteley 1983) that, for a general octahedron, when the leg lengths are not normalized,

$$\det J^* = [E_B \ E_A \ A \ C][E_A \ E_C \ C \ B][E_C \ E_B \ B \ A] \\ + [E_B \ E_A \ A \ B][E_A \ E_C \ C \ A][E_C \ E_B \ B \ C]. \quad (25)$$

Each bracket is simply the fourth order determinant of the coordinates of four finite points written homogeneously as  $(1, x_i, y_i, z_i)$ . The six points in (25) are the vertices of the octahedron. Each bracket is six times the volume of a tetrahedron the vertices of which are the four points. The dimension of  $\det J^*$  is  $(\text{volume})^3$ ; our previous definition of  $J$  requires  $\det J^*$  to be normalized with respect to the leg lengths.

Each bracket for the octahedron of Figure 2 (with the coordinates tabulated in (2))

yields the same absolute value,  $\left| \frac{\sqrt{3}}{2} abh \right|$ , and so

$$|\det J^*| = 2 \left( \frac{\sqrt{3}}{2} abh \right)^3 = \frac{3\sqrt{3}}{4} a^3 b^3 h^3. \quad (26)$$

Consider now the four-by-four array  $M$  of the homogeneous coordinates of the vertices of a tetrahedron formed by a pair of points 1 and 2 of the base triangle (i.e., any pair of  $E_A$ ,  $E_B$  and  $E_C$ ) for which  $z=-h$  and a pair of points 3 and 4 in the moving platform (i.e., any pair of  $A$ ,  $B$  and  $C$ ) the join of which is parallel to the base and a distance  $h$  from it.  $M$  can be written as

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 0 & 0 & -h & -h \end{bmatrix}. \quad (27)$$

When the moving platform translates parallel to the base it undergoes a shearing motion  $M \rightarrow M'$  where

$$M' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 + x & x_4 + x \\ y_1 & y_2 & y_3 + y & y_4 + y \\ 0 & 0 & -h & -h \end{bmatrix}. \quad (28)$$

The determinant of  $M'$  simplifies to

$$\det M' = -h \begin{vmatrix} 1 & 1 & 0 \\ x_1 & x_2 & x_3 - x_4 \\ y_1 & y_2 & y_3 - y_4 \end{vmatrix} \quad (29)$$

which is independent of extent of the shearing that is given by  $x$  and  $y$ . The result can be obtained directly from

$$M' = TM \quad (30)$$

where

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -x/h \\ 0 & 0 & 1 & -y/h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (31)$$

is the shear transformation, the determinant of which is 1. Then

$$\det M' = \det T \times \det M = \det M \quad (32)$$

It follows that each of the fourth-order determinants in (25) is invariant with the shear action and hence  $|\det J^*|$  is also invariant.

Now we can compute the induced six-by-six shear matrix that transforms the coordinates of the six legs by counting the 36 second-order determinants of  $T$  (equation (31)), namely

$$T_{6 \times 6} = \begin{bmatrix} 1 & 0 & -x/h & 0 & 0 & 0 \\ 0 & 1 & -y/h & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & x/h & y/h & 1 \end{bmatrix}, \quad (33)$$

and of course  $|\det T_{6 \times 6}| = 1$ .

Evaluating (25) with a rotation  $\theta$  about the vertical Z-axis yields

$$\begin{aligned} [E_B \ E_A \ A \ C] &= [E_A \ E_C \ C \ B] = [E_C \ E_B \ B \ A] \\ &= \left( \frac{1}{2} \sin \theta - \frac{\sqrt{3}}{2} \cos \theta \right) abh, \end{aligned} \quad (34)$$

$$\begin{aligned} [E_B \ E_A \ A \ B] &= [E_A \ E_C \ C \ A] = [E_C \ E_B \ B \ C] \\ &= \left( -\frac{1}{2} \sin \theta - \frac{\sqrt{3}}{2} \cos \theta \right) abh, \end{aligned} \quad (35)$$

and then

$$|\det J^*| = \frac{3\sqrt{3}}{4} a^3 b^3 h^3 \cos \theta. \quad (36)$$

The manipulator is at a special configuration when  $|\det J^*| = 0$ . Either one or more brackets (see (25)) in the first *and* second products vanish simultaneously, or the product of first three brackets is equal and opposite in sign to the product of the second

three brackets. This latter case corresponds to a  $90^\circ$  pure rotation about the Z-axis (see (34) and (35)).

Only when the four points of a bracket are coplanar is that bracket zero. Some of the special configurations described by Hunt and McAree (1998) correspond to two sets of four vertices of an octahedron being coplanar, one zero bracket being in the first product and one in the second. However, those special configurations for which no four vertices are coplanar correspond to the two products having equal absolute magnitudes but opposite signs. When the platform of Figure 2 is rotated through  $+60^\circ$  about the Z-axis all three brackets of one product are zero (three sets of four coplanar vertices), but no bracket of the other product is zero, and vice versa when angle rotated is  $-60^\circ$ . The configuration is, therefore, not special.

I do not pursue here the possibility of using (25) to find the maximum value of  $|\det J|$ , but certainly this alternative approach must yield identical results for our quality index.

## CHAPTER 4 FURTHER DEPARTURES FROM THE CENTRAL CONFIGURATION

I now derive an expression for  $|\det J|$  when the moving platform displaces in pure translation in the  $xy$ -plane. The coordinates for the center of the moving platform are now  $(x, y, 0)$  not  $(0, 0, 0)$  as in Figure 2,  $z$  remaining zero. From (4) the coordinates for each of the six legs can be expressed in the form  $\{L_i + x \quad M_i + y \quad N_i \quad ; \quad P_i \quad Q_i \quad R_i - y_j x + x_j y \quad (i=1, 2, \dots, 6)$  where  $L_i, M_i, N_i; P_i, Q_i$  and  $R_i$  are the columns of (12) and  $x_j$  and  $y_j$  ( $j=1, 2, 3$ ) are the coordinates of the base points  $E_A, E_B$  and  $E_C$ . For example the coordinates  $\hat{S}_1$  after a displacement  $(x, y)$  are

$$\hat{S}_1 \equiv \left[ \frac{a-b}{2} + x \quad \frac{a+b}{2\sqrt{3}} + y \quad z \quad ; \quad \frac{az}{2\sqrt{3}} \quad -\frac{az}{2} \quad \frac{ab}{2\sqrt{3}} + \frac{b}{2\sqrt{3}}x + \frac{b}{2}y \right]. \quad (37)$$

When (12) is set up using the new line coordinates the result is surprisingly simple. The numerator of  $\det J$  is invariant with translations of the moving platform in the  $xy$  plane i.e. shearing motions (see Chapter 3) and

$$|\det J| = \frac{3\sqrt{3} a^3 b^3 z^3}{4 l_1 l_2 l_3 l_4 l_5 l_6} \quad (38)$$

where

$$\begin{aligned}
l_1 &= \sqrt{\left(\frac{a-b}{2} + x\right)^2 + \left(\frac{a+b}{2\sqrt{3}} + y\right)^2 + z^2}, \quad l_2 = \sqrt{\left(\frac{a}{2} + x\right)^2 + \left(\frac{a-2b}{2\sqrt{3}} + y\right)^2 + z^2}, \\
l_3 &= \sqrt{\left(-\frac{a}{2} + x\right)^2 + \left(\frac{a-2b}{2\sqrt{3}} + y\right)^2 + z^2}, \quad l_4 = \sqrt{\left(\frac{b-a}{2} + x\right)^2 + \left(\frac{a+b}{2\sqrt{3}} + y\right)^2 + z^2}, \quad (39) \\
l_5 &= \sqrt{\left(\frac{b}{2} + x\right)^2 + \left(\frac{b-2a}{2\sqrt{3}} + y\right)^2 + z^2}, \quad l_6 = \sqrt{\left(-\frac{b}{2} + x\right)^2 + \left(\frac{b-2a}{2\sqrt{3}} + y\right)^2 + z^2},
\end{aligned}$$

and, from (38) and (17), the quality index is given by

$$\lambda = \frac{8\sqrt{3} \left(\sqrt{a^2 - ab + b^2}\right)^3 z^3}{9 l_1 l_2 l_3 l_4 l_5 l_6}. \quad (40)$$

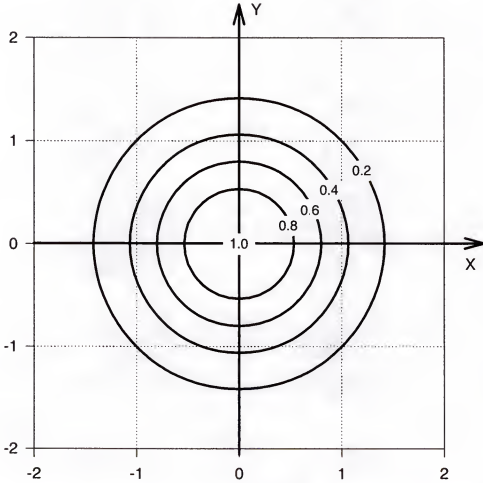


Figure 5 : Coplanar translation of platform from central location : contours of quality index

Figure 5 shows the contours of quality index as the platform of the octahedral manipulator is translated away from the central location parallel to the base plane taking  $h/h_m=1$ . The side of the platform is chosen as  $a=1$  and, from (20), the base side  $b=2$ .

The contours are labeled with values of constant quality index. When  $x$  or  $y$  is infinite,  $\lambda=0$ .

The contours are close to being concentric circles of various radii. Substituting  $x = r \cos \theta$  and  $y = r \sin \theta$  in (40) together with  $a = 1$ ,  $b = 2$  and  $z = 1$  yields

$$\lambda = \frac{8}{\sqrt{r^{12} + 6r^{10} + 21r^8 + (50 - 2\cos(6\theta))r^6 + 84r^4 + 96r^2 + 64}}. \quad (41)$$

It can easily be shown numerically that the term  $(50 - 2\cos(6\theta))r^6$  is negligible over the range  $0 \leq r \leq \infty$  since the upper and lower bounds for  $\lambda$  occurs when  $\cos 6\theta = \pm 1$  for which

$$\lambda_{upper} = \frac{8}{\sqrt{r^{12} + 6r^{10} + 21r^8 + 48r^6 + 84r^4 + 96r^2 + 64}}, \quad (42)$$

$$\lambda_{lower} = \frac{8}{\sqrt{r^{12} + 6r^{10} + 21r^8 + 52r^6 + 84r^4 + 96r^2 + 64}}. \quad (43)$$

The maximum deviation from a circle occurs when  $r = \sqrt{2}$ , for which  $\lambda_{upper} - \lambda_{lower} \approx 0.00203$ .

Figure 6 illustrates how the quality index varies as the platform is rotated away from its central location about a vertical Z-axis through its center, the legs being adjusted in length to keep the platform in the  $xy$  plane. The octahedron has the highest quality index,  $\lambda = 1$ , when  $\theta = 0^\circ$ ,  $\lambda = 0$  when  $\theta = \pm 90^\circ$ . It can, moreover, be shown that

platform rotation through  $\pm 90^\circ$  about any axis parallel to the Z-axis leads to a special configuration with  $\lambda = 0$  (see Hunt and McAre, 1998). Figure-7 illustrates the corresponding configuration of the platform.

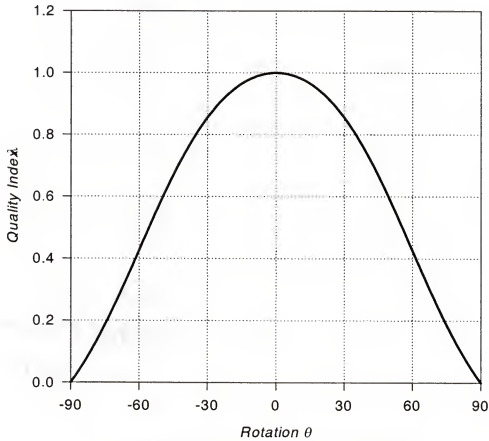


Figure 6 : Rotation of platform about Z-axis



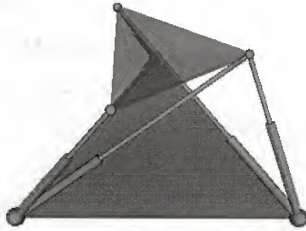


Figure 7 : A singularity configuration (Z axis rotation)

Rotations about lines in the  $xy$ -plane are linear combinations of rotations about the X- and Y-axis (see Figures 8 and 9 for the rotation component). Figure-10 shows a singularity configuration of the platform when the platform rotates  $60^\circ$  about X-axis.

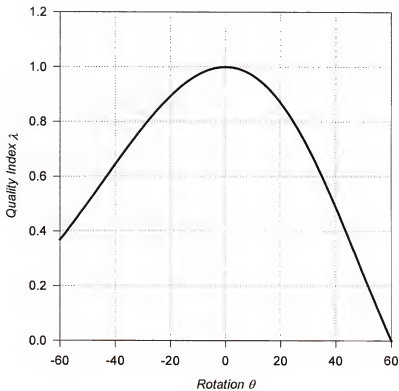


Figure 8 : Rotation of platform about X-axis

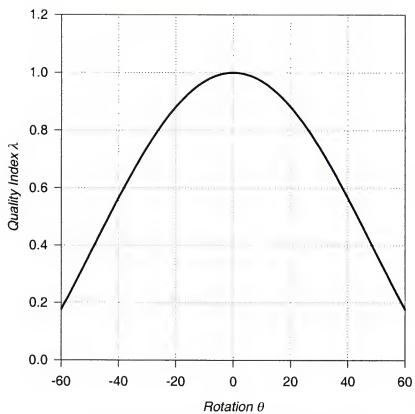


Figure 9 : Rotation of platform about Y-axis

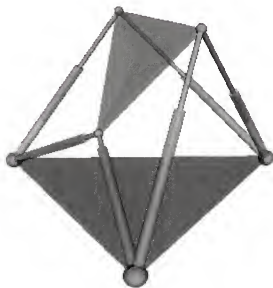


Figure 10 : A singularity configuration (X axis rotation)

CHAPTER 5  
DETERMINATION OF  $|\det J|_m$  FOR A 3-6 PLATFORM MANIPULATOR

Figure 11 illustrates the plan view of a 3-6 platform with a moving equilateral triangle of side  $a$  which is parallel to a hexagonal base with three pair of joints  $E_{A1,A2}$ ,  $E_{B1,B2}$  and  $E_{C1,C2}$ . Each pair of joints is separated by distance  $\beta b$  for which  $0 \leq \beta \leq 1$  and the moving platform is raised to distance  $h$ . Each of the six legs  $E_{A1}A$ ,  $E_{A2}A$ ,  $E_{B1}B$ ,  $E_{B2}B$ ,  $E_{C1}C$  and  $E_{C2}C$  are  $S$ - $P$ - $S$  connector chains.

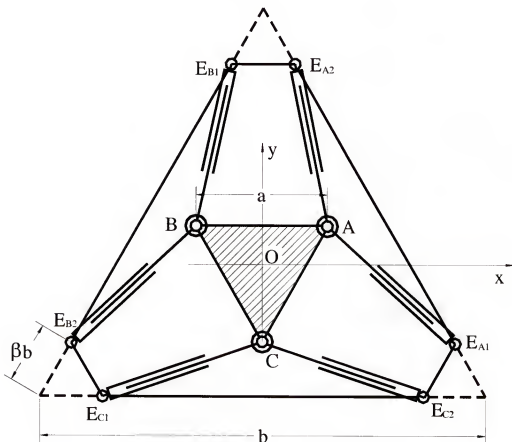


Figure 11 : Spatial 3-6 platform device (plan view)

Now the determinant of the Plücker coordinates of the six legs is used to determine a maximum value for  $\det J$ . First, the coordinates of the points  $A, B, C, E_{A1},$

$E_{A2}, E_{B1}, E_{B2}, E_{C1}$  and  $E_{C2}$  are determined referenced to the origin  $O$  located at the center of the base platform as illustrated.

$$\begin{aligned}
 & A \begin{pmatrix} \frac{a}{2} & \frac{a}{2\sqrt{3}} & h \end{pmatrix} \quad B \begin{pmatrix} -\frac{a}{2} & \frac{a}{2\sqrt{3}} & h \end{pmatrix} \quad C \begin{pmatrix} 0 & -\frac{a}{\sqrt{3}} & h \end{pmatrix} \\
 & E_{A1} \begin{pmatrix} \frac{b(1-\beta)}{2} & \frac{b(3\beta-1)}{2} & 0 \end{pmatrix} \quad E_{A2} \begin{pmatrix} \frac{\beta b}{2} & \frac{b(2-3\beta)}{2\sqrt{3}} & 0 \end{pmatrix} \\
 & E_{B1} \begin{pmatrix} -\frac{\beta b}{2} & \frac{b(2-3\beta)}{2\sqrt{3}} & 0 \end{pmatrix} \quad E_{B2} \begin{pmatrix} \frac{b(\beta-1)}{2} & \frac{b(3\beta-1)}{2\sqrt{3}} & 0 \end{pmatrix} \\
 & E_{C1} \begin{pmatrix} \frac{b(2\beta-1)}{2} & -\frac{b}{2\sqrt{3}} & 0 \end{pmatrix} \quad E_{C2} \begin{pmatrix} \frac{b(1-2\beta)}{2} & -\frac{b}{2\sqrt{3}} & 0 \end{pmatrix} \quad (44)
 \end{aligned}$$

and the corresponding line coordinates are

$$\hat{S}_1 = \left[ \frac{a-b(1-\beta)}{2} \quad \frac{a+b(1-3\beta)}{2\sqrt{3}} \quad h ; \quad \frac{bh(3\beta-1)}{2\sqrt{3}} \quad \frac{bh(\beta-1)}{2} \quad \frac{ab(1-2\beta)}{2\sqrt{3}} \right] \quad (45)$$

$$\hat{S}_2 = \left[ \frac{a-\beta b}{2} \quad \frac{a-b(2-3\beta)}{2\sqrt{3}} \quad h ; \quad \frac{bh(2-3\beta)}{2\sqrt{3}} \quad -\frac{\beta b h}{2} \quad \frac{ab(2\beta-1)}{2\sqrt{3}} \right] \quad (46)$$

$$\hat{S}_3 = \left[ \frac{\beta b - a}{2} \quad \frac{a-b(2-3\beta)}{2\sqrt{3}} \quad h ; \quad \frac{bh(2-3\beta)}{2\sqrt{3}} \quad \frac{\beta b h}{2} \quad \frac{ab(1-2\beta)}{2\sqrt{3}} \right] \quad (47)$$

$$\hat{S}_4 = \left[ \frac{b(1-\beta)-a}{2} \quad \frac{a+b(1-3\beta)}{2\sqrt{3}} \quad h ; \quad \frac{bh(3\beta-1)}{2\sqrt{3}} \quad \frac{bh(1-\beta)}{2} \quad \frac{ab(2\beta-1)}{2\sqrt{3}} \right] \quad (48)$$

$$\hat{S}_5 = \left[ \frac{b(1-2\beta)}{2} \quad \frac{b-2a}{2\sqrt{3}} \quad h ; \quad -\frac{bh}{2\sqrt{3}} \quad \frac{bh(1-2\beta)}{2} \quad \frac{ab(1-2\beta)}{2\sqrt{3}} \right] \quad (49)$$

$$\hat{S}_6 = \left[ \frac{b(2\beta-1)}{2} \quad \frac{b-2a}{2\sqrt{3}} \quad h ; \quad -\frac{bh}{2\sqrt{3}} \quad \frac{bh(2\beta-1)}{2} \quad \frac{ab(2\beta-1)}{2\sqrt{3}} \right] \quad (50)$$

From (45) to (50),

$$\det J = \frac{1}{l^6} \begin{vmatrix} \frac{a-b(1-\beta)}{2} & \frac{a-\beta b}{2} & \frac{\beta b-a}{2} & \frac{b(1-\beta)-a}{2} & \frac{b(1-2\beta)}{2} & \frac{b(2\beta-1)}{2} \\ \frac{a+b(1-3\beta)}{2\sqrt{3}} & \frac{a-b(2-3\beta)}{2\sqrt{3}} & \frac{a-b(2-3\beta)}{2\sqrt{3}} & \frac{a+b(1-3\beta)}{2\sqrt{3}} & \frac{b-2a}{2\sqrt{3}} & \frac{b-2a}{2\sqrt{3}} \\ h & h & h & h & h & h \\ \frac{bh(3\beta-1)}{2\sqrt{3}} & \frac{bh(2-3\beta)}{2\sqrt{3}} & \frac{bh(2-3\beta)}{2\sqrt{3}} & \frac{bh(3\beta-1)}{2\sqrt{3}} & -\frac{bh}{2\sqrt{3}} & -\frac{bh}{2\sqrt{3}} \\ \frac{bh(\beta-1)}{2} & -\frac{\beta b h}{2} & \frac{\beta b h}{2} & \frac{bh(1-\beta)}{2} & \frac{bh(1-2\beta)}{2} & \frac{bh(2\beta-1)}{2} \\ \frac{ab(1-2\beta)}{2\sqrt{3}} & \frac{ab(2\beta-1)}{2\sqrt{3}} & \frac{ab(1-2\beta)}{2\sqrt{3}} & \frac{ab(2\beta-1)}{2\sqrt{3}} & \frac{ab(1-2\beta)}{2\sqrt{3}} & \frac{ab(2\beta-1)}{2\sqrt{3}} \end{vmatrix} \quad (51)$$

where  $l = E_{A1}A = E_{A2}A = E_{B1}B = E_{B2}B = E_{C1}C = E_{C2}C$  and for any leg

$$l = \sqrt{L^2 + M^2 + N^2} = \sqrt{\frac{1}{3}(a^2 - ab + b^2(3\beta^2 - 3\beta + 1) + 3h^2)} \quad (52)$$

Expanding (51) yields

$$\det J = \frac{81\sqrt{3}a^3b^3h^3(2\beta-1)^3}{4(a^2 - ab + b^2(3\beta^2 - 3\beta + 1) + 3h^2)^3} \quad (53)$$

It is clear that when  $\beta = \frac{1}{2}$ , the platform is degenerated. Each pair of legs,  $E_{A1}A$  and  $E_{A2}A$ ,  $E_{B1}B$  and  $E_{B2}B$ , and  $E_{C1}C$  and  $E_{C2}C$  are collinear and  $\det J = 0$ .

Differentiating with respect to  $h$ ,  $\det J_m$  occurs when

$$h = \sqrt{\frac{1}{3}(a^2 - ab + b^2(3\beta^2 - 3\beta + 1))} \quad (54)$$

Substituting (54) into (53) yields

$$\det J_m = \frac{27a^3b^3(2\beta - 1)^3}{32(a^2 - ab + b^2(3\beta^2 - 3\beta + 1))^{\frac{3}{2}}} \quad (55)$$

Substituting  $b = \gamma a$  into (55) and dividing above and below by  $\gamma$  yields

$$\det J_m = \frac{27a^3(2\beta - 1)^3}{32\left(\frac{1}{\gamma^2} - \frac{1}{\gamma} + (3\beta^2 - 3\beta + 1)\right)^{\frac{3}{2}}} \quad (56)$$

The absolute maximum value of  $\det J_m$  is obtain by taking the derivative of (56) with respect to  $\gamma$  which yields

$$\frac{1}{\gamma^2} \left(1 - \frac{2}{\gamma}\right) = 0 \quad (57)$$

and

$$\gamma = \frac{b}{a} = 2 \quad (58)$$

It is interesting to note that the moving and base platform ratio  $\gamma$  which is required to define  $\det J_{am}$  is independent to a value of  $\beta$ . Further from (56) when  $\gamma = 2$ ,

$$\det J_{am} = \frac{3\sqrt{3}}{4} a^3 \quad (59)$$

which is identical to the result for the octahedron platform device with moving platform sides  $a$ , base sides  $2a$  and height  $a$ . It is clear that  $\det J_{am}$  is not dependent to the value of  $\beta$ . This special 3-6 platform with  $h = a\sqrt{3\beta^2 - 3\beta + 1}$  (see (54)) is shown in the optimal configuration in Figure 12.

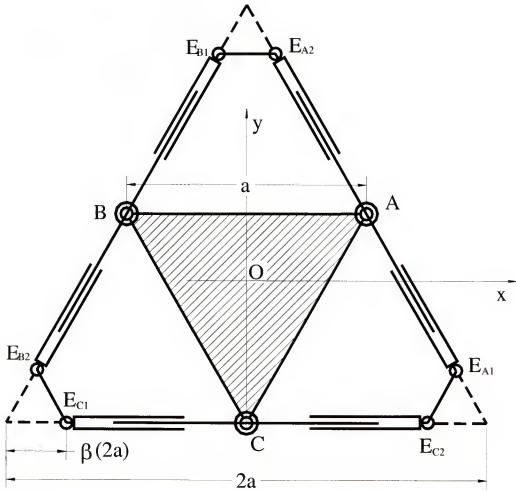


Figure 12 : Spatial 3-6 platform in optimal configuration

CHAPTER 6  
DETERMINATION OF  $|\det J|_m$  FOR A 6-6 PLATFORM MANIPULATOR

Figure 13 illustrates the plan view of a 6-6 platform device. The moving platform joints are separated by  $\alpha a$  where  $0 \leq \alpha \leq 1$  and base joints are also separated by  $\beta b$  where  $0 \leq \beta \leq 1$  from their original double joints. The platform is raised to a distance  $h$ . Each of the six legs  $E_{A1}A_1$ ,  $E_{A2}A_2$ ,  $E_{B1}B_1$ ,  $E_{B2}B_2$ ,  $E_{C1}C_1$  and  $E_{C2}C_2$  are *S-P-S* connector chains.

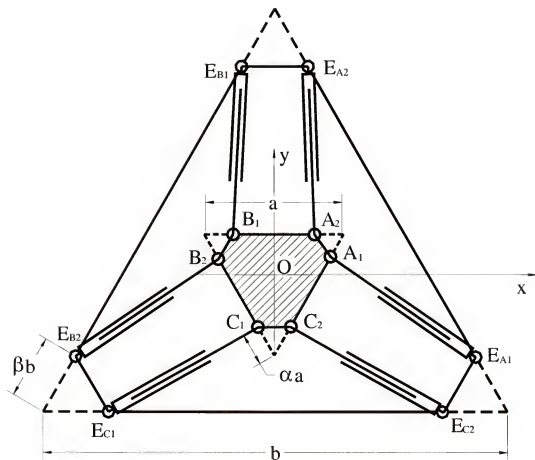


Figure 13 : Spatial 6-6 platform device (plan view)



The coordinates of the points  $A_1, B_1, C_1, A_2, B_2, C_2, E_{A1}, E_{A2}, E_{B1}, E_{B2}, E_{C1}$  and  $E_{C2}$  are determined with  $O$  located at the center of the base platform to determine the Plücker coordinates of the six legs.

$$\begin{aligned}
 A_1 & \left( \frac{a(1-\alpha)}{2} \quad \frac{a(1-3\alpha)}{2\sqrt{3}} \quad h \right) & A_2 & \left( \frac{a(1-2\alpha)}{2} \quad \frac{a}{2\sqrt{3}} \quad h \right) \\
 B_1 & \left( \frac{a(2\alpha-1)}{2} \quad \frac{a}{2\sqrt{3}} \quad h \right) & B_2 & \left( \frac{a(\alpha-1)}{2} \quad \frac{a(1-3\alpha)}{2\sqrt{3}} \quad h \right) \\
 C_1 & \left( -\frac{\alpha a}{2} \quad \frac{a(3\alpha-2)}{2\sqrt{3}} \quad h \right) & C_2 & \left( \frac{\alpha a}{2} \quad \frac{a(3\alpha-2)}{2\sqrt{3}} \quad h \right) \\
 E_{A1} & \left( \frac{b(1-\beta)}{2} \quad \frac{b(3\beta-1)}{2\sqrt{3}} \quad 0 \right) & E_{A2} & \left( \frac{\beta b}{2} \quad \frac{b(2-3\beta)}{2\sqrt{3}} \quad 0 \right) \\
 E_{B1} & \left( -\frac{\beta b}{2} \quad \frac{b(2-3\beta)}{2\sqrt{3}} \quad 0 \right) & E_{B2} & \left( \frac{b(\beta-1)}{2} \quad \frac{b(3\beta-1)}{2\sqrt{3}} \quad 0 \right) \\
 E_{C1} & \left( \frac{b(2\beta-1)}{2} \quad -\frac{b}{2\sqrt{3}} \quad 0 \right) & E_{C2} & \left( \frac{b(1-2\beta)}{2} \quad -\frac{b}{2\sqrt{3}} \quad 0 \right) \quad (60)
 \end{aligned}$$

By following the same steps (3) through (12), all the leg lengths, which are the same for all six, and the determinant of Jacobian matrix of six legs are,

$$l = \sqrt{\frac{1}{3} \left( a^2 (3\alpha^2 - 3\alpha + 1) + ab(3\alpha\beta - 1) + b^2 (3\beta^2 - 3\beta + 1) + 3h^2 \right)} \quad (61)$$

and

$$\det J = \frac{81\sqrt{3}a^3b^3h^3(3\alpha\beta - 2\alpha - 2\beta + 1)^3}{4(a^2(3\alpha^2 - 3\alpha + 1) + ab(3\alpha\beta - 1) + b^2(3\beta^2 - 3\beta + 1) + 3h^2)^3} \quad (62)$$

It is interesting to note that  $\det J = 0$  when  $3\alpha\beta - 2\alpha - 2\beta + 1 = 0$ . This relationship always yields similar base and moving platform and some of these cases are illustrated in Figure 14,  $\alpha = \beta = \frac{1}{3}$ ,  $\alpha = \frac{1}{4}$ ,  $\beta = \frac{2}{5}$  and  $\alpha = 0$ ,  $\beta = \frac{1}{2}$ . All such platforms have finite mobility when all six actuators are locked. All the six connector legs lie on a regulus and the platform has three degree of freedom since these connector lines belong to a three system. Such mechanisms belong to the Bricard class which have full-cycle mobility. The plot of the function is illustrated by Figure 14.

Differentiating (62) with respect to  $h$ ,  $\det J_m$  occurs when

$$h = \sqrt{\frac{1}{3}(a^2(3\alpha^2 - 3\alpha + 1) + ab(3\alpha\beta - 1) + b^2(3\beta^2 - 3\beta + 1))} \quad (63)$$

and then substituting (63) into (62) yields

$$\det J_m = \frac{27a^3b^3(3\alpha\beta - 2\alpha - 2\beta + 1)^3}{32(a^2(3\alpha^2 - 3\alpha + 1) + ab(3\alpha\beta - 1) + b^2(3\beta^2 - 3\beta + 1))^{\frac{3}{2}}} \quad (64)$$

Substituting  $b = \gamma a$  into (64) and dividing above and below by  $\gamma$  yields

$$\det J_m = \frac{27a^3(3\alpha\beta - 2\alpha - 2\beta + 1)^3}{32\left(\frac{1}{\gamma^2}(3\alpha^2 - 3\alpha + 1) + \frac{1}{\gamma}(3\alpha\beta - 1) + (3\beta^2 - 3\beta + 1)\right)^{\frac{3}{2}}} \quad (65)$$

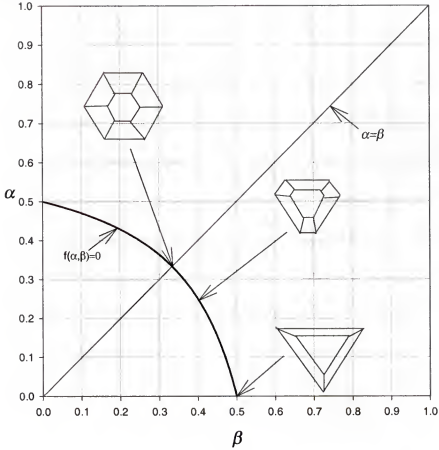


Figure 14 : Plot of  $f(\alpha, \beta) = 3\alpha\beta - 2\alpha - 2\beta + 1 = 0$  and configurations for  $\det J = 0$

The absolute maximum value of  $\det J_m$  can be determined by taking the derivative of (65) with respect to  $\gamma$  which yields

$$-\frac{1}{\gamma^2} \left( \frac{2}{\gamma} (3\alpha^2 - 3\alpha + 1) + 3\alpha\beta - 1 \right) = 0 \quad (66)$$

and therefore

$$\gamma = \frac{b}{a} = \frac{-6\alpha^2 + 6\alpha - 2}{3\alpha\beta - 1} \quad (67)$$

The ratio  $\gamma$  is now dependent to the values of both  $\alpha$  and  $\beta$ . Further when  $\alpha = 0$ , the result is the same as 3-6 platform ( $\gamma = 2$ ) and when  $\beta = 0$ ,  $\gamma = 6\alpha^2 - 6\alpha + 2$ .

Substituting (67) into (65) yields

$$\det J_{am} = \frac{3\sqrt{3}}{4} a^3 (3\alpha^2 - 3\alpha + 1)^{\frac{3}{2}} \quad (68)$$

It is interesting to note that  $\det J_{am}$  is dependent only to the value of  $\alpha$ . It is independent to a value of  $\beta$ . When  $\alpha = 0$  or  $1$ ,  $\det J_{am}$  has maximum value of  $\frac{3\sqrt{3}}{4} a^3$ ,

and when  $\alpha = \frac{1}{2}$ ,  $\det J_{am}$  becomes minimal,  $\det J_{am} = \frac{3\sqrt{3}}{32} a^3$ .

This 6-6 platform with  $\alpha = \frac{1}{4}$ ,  $\beta = \frac{1}{6}$  and  $\gamma = 1$  (or  $b = a$ ) is shown in the optimal configuration in Figure 15 and for which, from (63),  $h = \frac{\sqrt{7}}{12} a$ . Its absolute maximum of

$\det J_m$  is  $\frac{21\sqrt{21}}{256} a^3$  from (68).

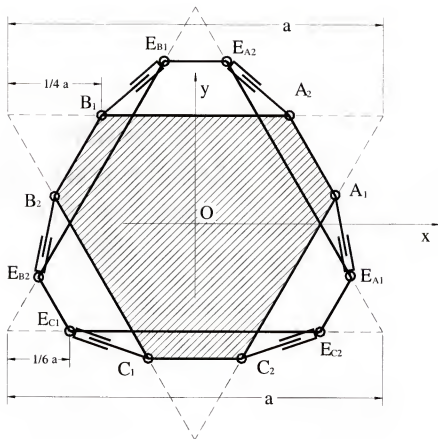


Figure 15 : Spatial 6-6 platform in optimal configuration  
 $(\alpha = 1/4, \beta = 1/6, b = a, h = \sqrt{7}a/12)$

## CHAPTER 7

### COMPATABILITY OF THE RESULTS OF THE 3-3, 3-6, 6-6 PLATFORMS

The maximum values of determinants of Jacobian matrix which is used for quality index for a 3-3 octahedron, a spatial 3-6 and a spatial 6-6 platform device have been obtained.

For the 3-3 octahedral platform device and the 3-6 device, the absolute maximum value of the determinant of  $J$ ,  $\det J_{am} = \frac{3\sqrt{3}}{4}a^3$ , which is the volume of an octahedron with equilateral triangular moving platform with side  $a$  and an equilateral base side  $2a$ . The value of  $\det J_{am}$  for a 3-6 device with the same moving platform  $ABC$  (side  $a$ ) is the same,  $\det J_{am} = \frac{3\sqrt{3}}{4}a^3$ . Figure 16 illustrates the compatibility of these two results. It can be observed that as the distance between the pairs of separation points of the double ball and socket joints  $E_A$ ,  $E_B$  and  $E_C$  of the original octahedron are increased the value of  $h$  is decreased from  $h = a$  to obtain the configuration for which  $\det J = \det J_{am}$ . These results are analogous to the planar case (see Figure 1) for which  $\det J_{am} = \frac{3l}{2}$  regardless of the location of base pivots.

However, for the 6-6 device, the result is different than those for the 3-3 octahedral or the 3-6 devices. The reason is that there is a reduction of overall size of the moving platform. Because the absolute maximum value of the determinant of the

Jacobian matrix solely relies on the dimension of the moving platform, once the double joints of the moving platform are separated, the dimension of the platform changes. For example (see Figure 17), when  $\alpha = \frac{1}{2}$ , the pair of separated joints of the moving platform recombine in a different sequence to form double joints again. The joints  $A_2B_1$ ,  $B_2C_1$  and  $C_2A_1$  now combine to form double joints and hence a moving triangle with sides  $\frac{a}{2}$ . The 6-6 platform has thus reduced to a new 3-6 platform. With this configuration, from (68),

$$\det J_{am} = \frac{3\sqrt{3}}{32} a^3 \text{ and also from (59) with } a = \frac{a}{2}, \det J_{am} = \frac{3\sqrt{3}}{4} \left(\frac{a}{2}\right)^3 = \frac{3\sqrt{3}}{32} a^3.$$

Now it is clear that the absolute maximum values of  $\det J$  for octahedral 3-3 platform device and spatial 3-6 device which both have an equilateral triangular moving platform are the same,  $\det J_{am} = \frac{3\sqrt{3}}{4} a^3$ . For the spatial 6-6 device, the absolute maximum value of  $\det J$  depends on the separation of the joints which changes the dimension (or shape) of the moving platform and also when the shape of the moving platform is getting close to a triangle, the value of  $\det J_{am}$  increases. But it does not exceed  $\frac{3\sqrt{3}}{4} a^3$ , which is the value for the 3-3 and 3-6 devices.

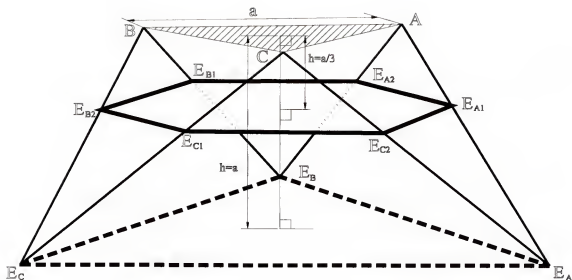


Figure 16 : Compatibility between the 3-3 octahedral platform and the 3-6 platform

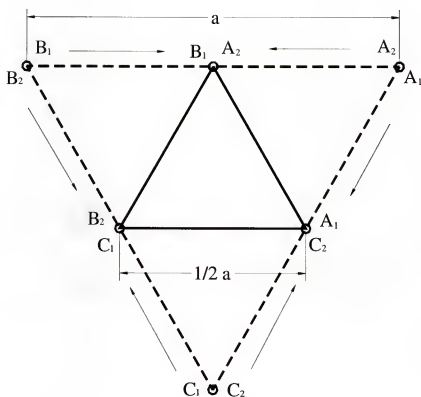


Figure 17 : Reduction of the size of the moving platform when  $\alpha = 1/2$



## CHAPTER 8

### OTHER ALTERNATIVE JOINT SEPARATIONS

Quality indexes for some simple spatial platform devices have been analyzed and they have given ideas on how to design a platform. Still, there is a lot more to discover about singularities and configurations that cause them. All the platforms analyzed so far, have been derived from an octahedron platform because an octahedron is one of the strongest polyhedrons that have triangular faces such as the tetrahedron and icosahedron. There are more ways to separate double joints other than those discussed in this dissertation. As we pointed out with the 6-6 platform device, certain arrangements of six joints on the moving and base platform cause certain degenerate conditions. They are not always easy to find but at least some of them can be determined using the quality index analysis technique.

#### Joint Separation Toward the Center of the Platform

Figure-18 illustrates the plan view of a platform with one of each double joints moved toward the center of the platform. These joints are separated by a distance  $\rho d_a$  from the moving platform joints and  $\rho d_b$  from the base joints where  $d_a$  is the distance from the joints of the moving platform to the center of the moving platform and  $d_b$  is the distance from the base joints to the base center. The value  $\rho$  varies from 0 to 1. The moving platform is raised to distance  $h$  from the base. Each of the six legs  $E_{A1}A_1$ ,  $E_{A2}A_2$ ,  $E_{B1}B_1$ ,  $E_{B2}B_2$ ,  $E_{C1}C_1$  and  $E_{C2}C_2$  are *S-P-S* connector chains.

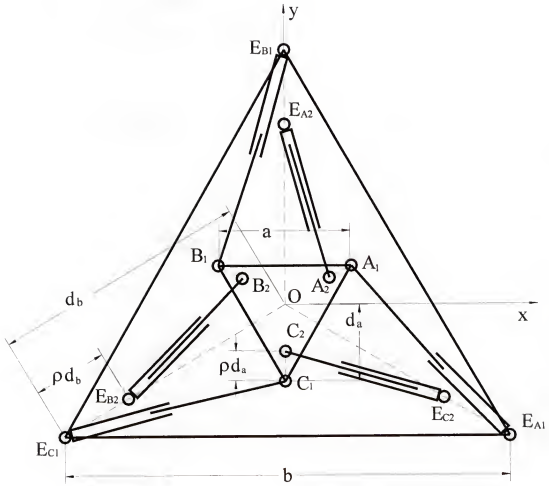


Figure 18 : Platform with the joints separated toward the center

The coordinates of the points  $A_1$ ,  $B_1$ ,  $C_1$ ,  $A_2$ ,  $B_2$ ,  $C_2$ ,  $E_{A1}$ ,  $E_{A2}$ ,  $E_{B1}$ ,  $E_{B2}$ ,  $E_{C1}$  and  $E_{C2}$  are determined with  $O$  located at the center of the base platform to determine the Plücker coordinates of the six legs. Because the pairs of the connector legs of the platform are not symmetrical about the lines  $OE_{A1}$ ,  $OE_{B1}$  and  $OE_{C1}$ , the platform must be rotated about the  $Z$ -axis to reach the optimal configuration. This rotation factor  $\theta$  is added to the top platform joint coordinates and they are,

$$\begin{aligned}
A_1 & \left( \frac{a \cos \theta}{2} - \frac{a \sin \theta}{2\sqrt{3}} \quad \frac{a \sin \theta}{2} + \frac{a \cos \theta}{2\sqrt{3}} \quad h \right) \\
A_2 & \left( \frac{a(1-\rho) \cos \theta}{2} - \frac{a(1-\rho) \sin \theta}{2\sqrt{3}} \quad \frac{a(1-\rho) \sin \theta}{2} + \frac{a(1-\rho) \cos \theta}{2\sqrt{3}} \quad h \right) \\
B_1 & \left( -\frac{a \cos \theta}{2} - \frac{a \sin \theta}{2\sqrt{3}} \quad -\frac{a \sin \theta}{2} + \frac{a \cos \theta}{2\sqrt{3}} \quad h \right) \\
B_2 & \left( -\frac{a(1-\rho) \cos \theta}{2} - \frac{a(1-\rho) \sin \theta}{2\sqrt{3}} \quad -\frac{a(1-\rho) \sin \theta}{2} + \frac{a(1-\rho) \cos \theta}{2\sqrt{3}} \quad h \right) \\
C_1 & \left( \frac{a \sin \theta}{\sqrt{3}} \quad -\frac{a \cos \theta}{\sqrt{3}} \quad h \right) \\
C_2 & \left( \frac{a(1-\rho) \sin \theta}{\sqrt{3}} \quad -\frac{a(1-\rho) \cos \theta}{\sqrt{3}} \quad h \right) \\
E_{A1} & \left( \frac{b}{2} \quad -\frac{b}{2\sqrt{3}} \quad 0 \right) & E_{A2} & \left( 0 \quad \frac{b(1-\rho)}{\sqrt{3}} \quad 0 \right) \\
E_{B1} & \left( 0 \quad \frac{b}{\sqrt{3}} \quad 0 \right) & E_{B2} & \left( -\frac{b(1-\rho)}{2} \quad -\frac{b(1-\rho)}{2\sqrt{3}} \quad 0 \right) \\
E_{C1} & \left( -\frac{b}{2} \quad -\frac{b}{2\sqrt{3}} \quad 0 \right) & E_{C2} & \left( \frac{b(1-\rho)}{2} \quad -\frac{b(1-\rho)}{2\sqrt{3}} \quad 0 \right). \quad (69)
\end{aligned}$$

By following the same steps (3) through (12), the length of the three inner legs is,

$$l_i = \sqrt{\frac{1}{3} (3h^2 + a^2(1-\rho)^2 + b^2(1-\rho)^2 - ab(1-\rho)^2 \cos \theta - \sqrt{3}ab(1-\rho)^2 \sin \theta)} \quad (70)$$

and the length of the three outer legs is

$$l_o = \sqrt{\frac{1}{3} (a^2 - ab \cos \theta + \sqrt{3}ab \sin \theta + b^2 + 3h^2)} \quad (71)$$

The determinant of Jacobian matrix of the six legs is,

$$\det J = \frac{3a^3b^3h^3(1-\rho)^2(\sqrt{3}(\rho^2-2\rho+2)\cos\theta - (\rho^2-2\rho+2)\sin\theta)}{8l_1^3l_o^3} \quad (72)$$

It is clear that  $\det J = 0$  when  $\rho = 1$ . Differentiating (72) with respect to  $h$ ,  $\det J_m$  occurs when

$$h = \sqrt[4]{\frac{1}{9}(a^4(1-\rho)^2 + a^2b^2(1-\rho)^2 + 2a^2b^2(1-\rho)^2 \cos 2\theta - 2ab(a^2+b^2)(1-\rho)^2 \cos \theta + b^4(1-\rho)^2)} \quad (73)$$

Unfortunately, it is difficult to continue the investigation of  $\det J_m$  analytically because the equation of  $\det J_m$  becomes too complex. Now, the dimensions of the platform are substituted with  $a=1$  and  $b=2$  and, from (73),

$$h = \sqrt[4]{\frac{1}{9}(1-\rho)^2(21 + 8\cos 2\theta - 20\cos \theta)}.$$

Figure-19 illustrates the values of  $\det J_m$  versus the values of the separation factor  $\rho$ . When  $\rho=0$ ,  $\det J_m$  has the maximum value of 1.2990 or  $\frac{3\sqrt{3}}{4}$  which is the same as  $\det J_m$  of the 3x3 octahedral platform and when  $\rho=1$ , the  $\det J_m$  equals to zero where the platform is in a singularity.

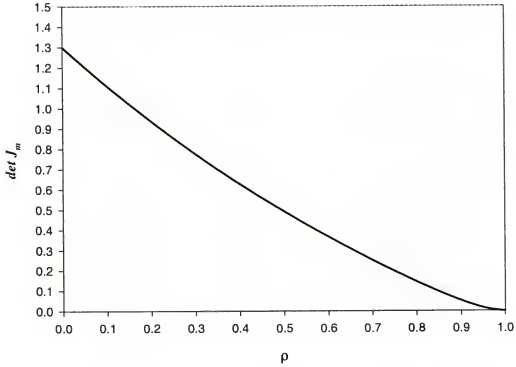


Figure 19 :  $\det J_m$  vs  $\rho$  (platform with joints separated toward the center)

Figures-20 and 21 illustrate the relations between  $\rho$  and  $\theta$  and between  $\rho$  and  $h$  when  $\det J_m$  occurs. For example, when  $\rho = 0.2$  the platform must be rotated by  $-3.7^\circ$  about Z-axis and raised by 0.8938 from the base to reach its optimum configuration. It is interesting to note that when the separation value  $\rho$  increases, the platform rotates more about the Z-axis (negative direction) and translates lower along the Z-axis from the optimal configuration of the 3x3 octahedral platform to reach its optimal configuration (see Figures-20 and 21).

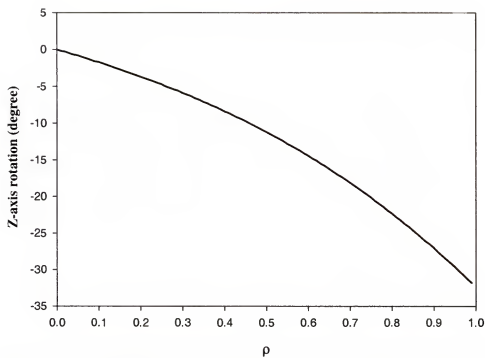


Figure 20 : Z-axis rotation vs  $\rho$  (platform with joints separated toward the center)

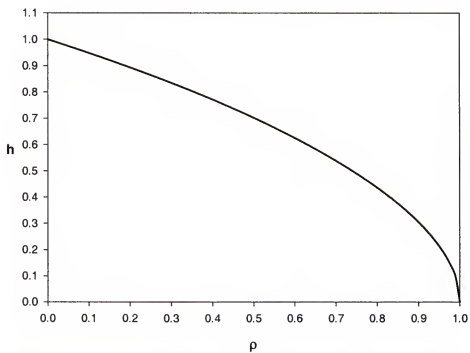


Figure 21 :  $h$  vs  $\rho$  (platform with joints separated toward the center)



The quality index varies in concentric circular shapes when the platform translates parallel to the x-y plane (see Figure-23) and has the optimal value of 1 at the center where  $x = 0$  and  $y = 0$ . The quality index increases as the platform is raised from the base and it becomes 1 when the platform reaches the height of 0.95 (see Figure-24). Once the platform reaches its optimal height, the quality index starts decreasing as the height increases.

A singularity occurs when the platform rotates  $-75^\circ$  and  $+105^\circ$  about the X-axis,  $\pm 90^\circ$  about the Y-axis and  $-85^\circ$  and  $+95^\circ$  about the Z-axis (see Figures-25 through 27).

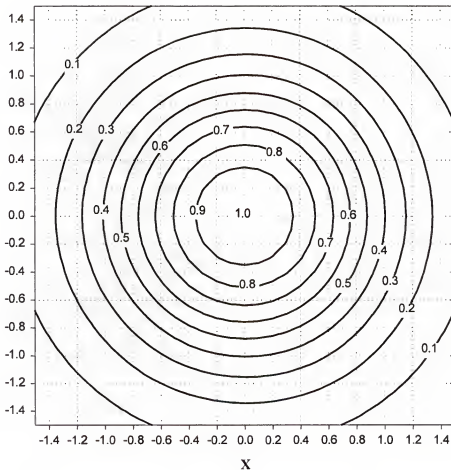


Figure 23 : Variation of quality index on x-y plane : contours of quality index (platform with joints separated toward the center)



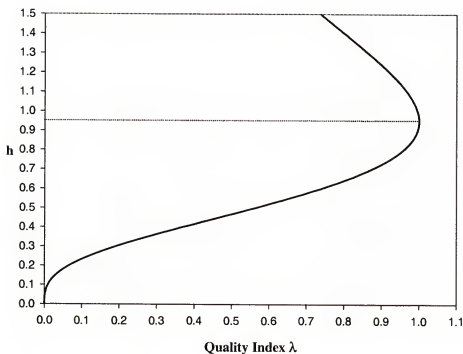


Figure 24 : Variation of quality index along Z-axis (platform with joints separated toward the center)

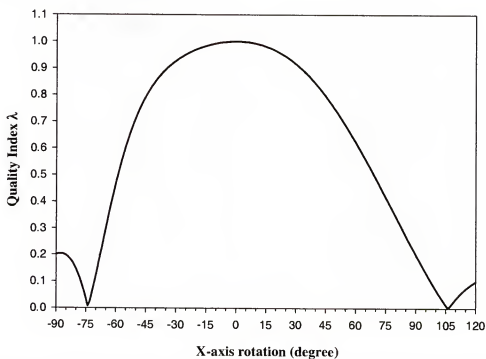


Figure 25 : Variation of quality index when platform rotates about X-axis (platform with joints separated toward the center)

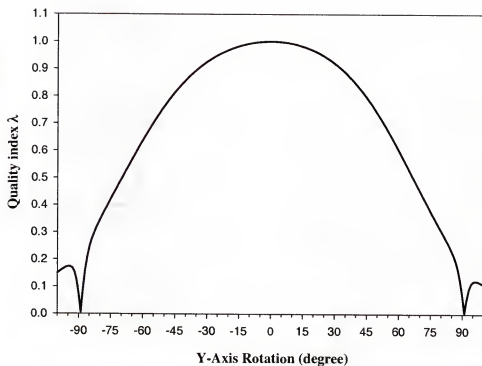


Figure 26 : Variation of quality index when platform rotates about Y-axis (platform with joints separated toward the center)

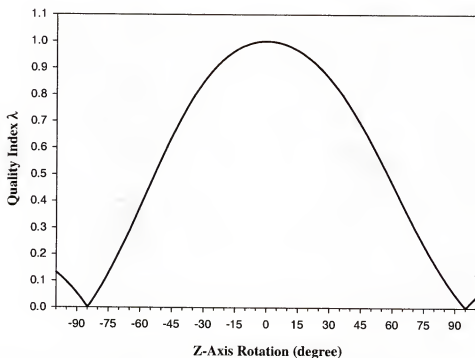


Figure 27 : Variation of quality index when platform rotates about Z-axis (platform with joints separated toward the center)

Joint Separation Along One Side of the Triangle

Figure-28 illustrates the plan view of a platform with one of each double joints moved along the triangle edges of the platform. These joints are separated by distance  $\rho b$  for the base joints and  $\rho a$  for the moving platform joints ( $0 \leq \rho \leq 1$ ). The moving platform is raised to distance  $h$ . The six legs connecting the outer base joints to the inner platform joints ( $E_A A_1$ ,  $E_B B_1$ ,  $E_C C_1$ ) and the the inner base joints to outer platform joints ( $E_{A2} A_2$ ,  $E_{B2} B_2$ ,  $E_{C2} C_2$ ) are *S-P-S* connector chains as before.

Because this platform also has non-symmetrical pairs of the connector legs, the rotation  $\theta$  of the top platform about the Z-axis is added to the top platform joint coordinates. All the joint coordinates are,

$$\begin{aligned}
 A_1 & \left( \frac{a(1-\rho)\cos\theta}{2} - \frac{a(1-3\rho)\sin\theta}{2\sqrt{3}} \quad \frac{a(1-\rho)\sin\theta}{2} + \frac{a(1-3\rho)\cos\theta}{2\sqrt{3}} \quad h \right) \\
 A_2 & \left( \frac{a\cos\theta}{2} - \frac{a\sin\theta}{2\sqrt{3}} \quad \frac{a\sin\theta}{2} + \frac{a\cos\theta}{2\sqrt{3}} \quad h \right) \\
 B_1 & \left( -\frac{a(1-2\rho)\cos\theta}{2} - \frac{a\sin\theta}{2\sqrt{3}} \quad -\frac{a(1-2\rho)\sin\theta}{2} + \frac{a\cos\theta}{2\sqrt{3}} \quad h \right) \\
 B_2 & \left( -\frac{a\cos\theta}{2} - \frac{a\sin\theta}{2\sqrt{3}} \quad -\frac{a\sin\theta}{2} + \frac{a\cos\theta}{2\sqrt{3}} \quad h \right)
 \end{aligned}$$

$$C_1 \left( -\frac{\rho a \cos \theta}{2} + \frac{a(2-3\rho) \sin \theta}{2\sqrt{3}} - \frac{\rho a \sin \theta}{2} - \frac{a(2-3\rho) \cos \theta}{2\sqrt{3}} \quad h \right)$$

$$C_2 \left( \frac{a \sin \theta}{\sqrt{3}} - \frac{a \cos \theta}{\sqrt{3}} \quad h \right)$$

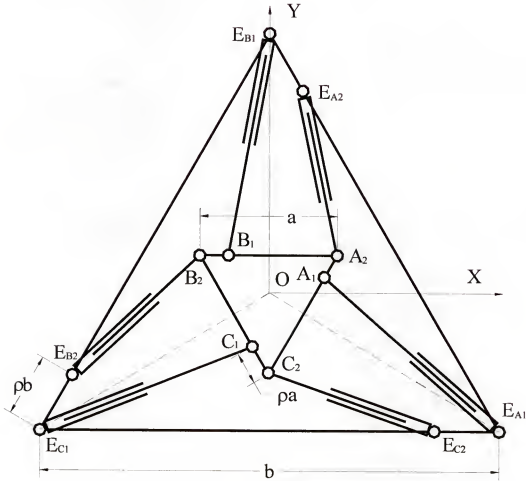


Figure 28 : Platform with joints separated along one edge of the triangles

$$E_{A1} \left( \frac{b}{2} \quad -\frac{b}{2\sqrt{3}} \quad 0 \right)$$

$$E_{A2} \left( \frac{\rho b}{2} \quad \frac{b(2-3\rho)}{2\sqrt{3}} \quad 0 \right)$$

$$\begin{aligned}
E_{B1} & \begin{pmatrix} 0 & \frac{b}{\sqrt{3}} & 0 \end{pmatrix} & E_{B2} & \begin{pmatrix} -\frac{b(1-\rho)}{2} & -\frac{b(1-3\rho)}{2\sqrt{3}} & 0 \end{pmatrix} \\
E_{C1} & \begin{pmatrix} -\frac{b}{2} & -\frac{b}{2\sqrt{3}} & 0 \end{pmatrix} & E_{C2} & \begin{pmatrix} \frac{b(1-2\rho)}{2} & -\frac{b}{2\sqrt{3}} & 0 \end{pmatrix}
\end{aligned} \quad (74)$$

By following the same steps (3) through (12), the length of the three legs of  $E_{A1}A_1$ ,  $E_{B1}B_1$  and  $E_{C1}C_1$  is,

$$l_{1,3,5} = \sqrt{\frac{1}{3} \left( 3h^2 + a^2(3\rho^2 - 3\rho + 1) - (\cos\theta + \sqrt{3}(2\rho - 1)\sin\theta)ab + b^2 \right)} \quad (75)$$

and the length of the three legs of  $E_{A2}A_2$ ,  $E_{B2}B_2$  and  $E_{C2}C_2$  is,

$$l_{2,4,6} = \sqrt{\frac{1}{3} \left( 3h^2 + a^2 - (\cos\theta - \sqrt{3}(2\rho - 1)\sin\theta)ab + b^2(3\rho^2 - 3\rho + 1) \right)} \quad (76)$$

The determinant of Jacobian matrix of the six legs is,

$$\det J = \frac{3\sqrt{3}a^3b^3h^3(6\rho^5 - 15\rho^4 + 20\rho^3 - 15\rho^2 + 6\rho - 1)\cos\theta}{4l_{1,3,5}^3 l_{2,4,6}^3} \quad (77)$$

It is clear that, from the factor  $(6\rho^5 - 15\rho^4 + 20\rho^3 - 15\rho^2 + 6\rho - 1)$  in (77), when  $\rho = 1/2$ , the platform is in a singularity. And also when the platform rotates  $90^\circ$  about the Z-axis, the platform is also in a singularity.

Using the platform dimensions of  $a=1$  and  $b=2$ , the relations between  $\det J_m$  and  $\rho$ , between  $\theta$  and  $\rho$  and between  $h$  and  $\rho$  are illustrated in Figures-29 through 31.

The  $\det J_m$  has the maximum value of 1.2990 or  $\frac{3\sqrt{3}}{4}$  when  $\rho=0$  and after that the

$\det J_m$  decreases as  $\rho$  increases. As previous mentioned (see (77)), when  $\rho=0.5$ ,  $\det J_m$  equals to zero and the platform is in a singularity everywhere (see Figure-29).

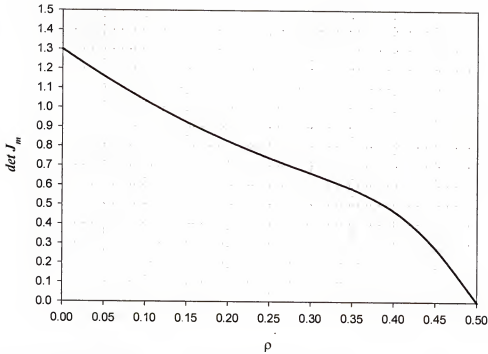


Figure 29 :  $\det J_m$  vs  $\rho$  (platform with joint separated along one edge of the triangle)

It is interesting to note that the angle of rotation about Z-axis of the top platform increases to reach the optimal configuration as  $\rho$  increases and when  $\rho$  reaches at the value of 0.32, the angle of rotation starts decreasing. The height  $h$  decreases as  $\rho$  increases.

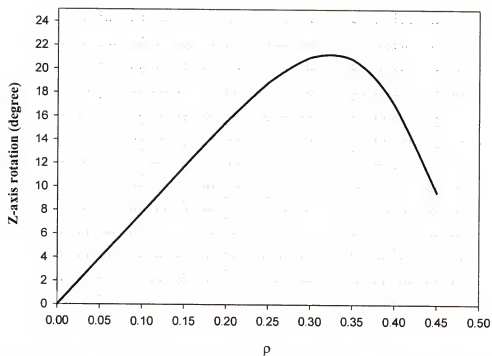


Figure 30 : Z-axis rotation vs  $\rho$  (platform with joint separated along one edge of the triangle)

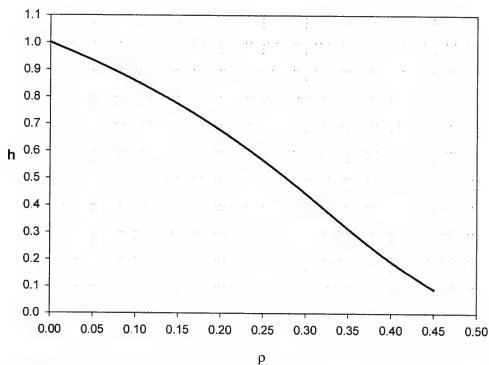


Figure 31 :  $h$  vs  $\rho$  (platform with joint separated along one edge of the triangle)

Figure-32 illustrates the platform with  $a=1$ ,  $b=2$  and  $\rho = 0.1$ . It has the maximum determinant value of 1.0347 when the platform rotates  $7.73^\circ$  about Z-axis and raised 0.86 from the base.

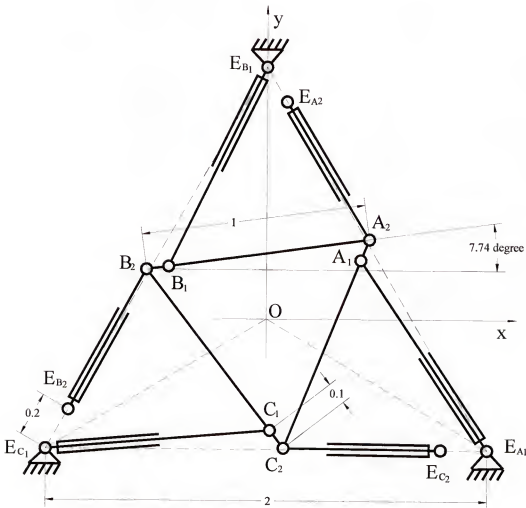


Figure 32 : Optimal configuration of platform with joint separated along one edge of the triangle

As it is mentioned earlier, moving the platform away from the optimal configuration makes the quality index decrease. Figures 33 through 37 illustrates the quality index variations when the platform departs from its optimal configuration.



The quality index varies in distorted concentric circles which are symmetrical to the dotted lines when the platform translates parallel to the x-y plane and has the optimal value of 1 at the center where  $x = 0$  and  $y = 0$  (see Figure-33). The quality index increases as the platform is raised from the base and it has the highest value of 1 when the platform reaches the height of 0.86 (see Figure-34). Once the platform reaches its optimal height, the quality index starts decreasing as the height increases.

A singularity occurs when the platform rotates  $-71^\circ$  and  $+108^\circ$  about X-axis,  $-95^\circ$  and  $+85^\circ$  about Y-axis, and  $-98^\circ$  and  $+82^\circ$  about Z-axis (see Figure-35 through 37).

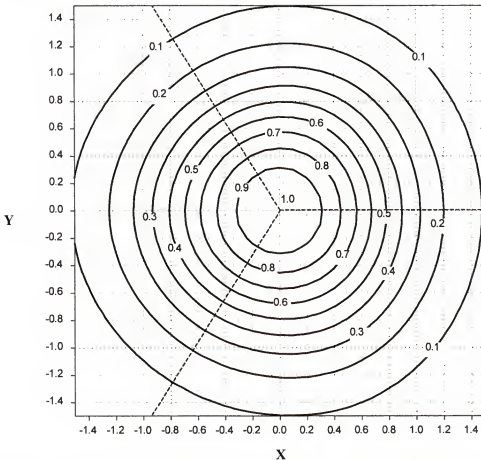


Figure 33 : Variation of quality index on x-y plane : contours of quality index (platform with joint separated along one edge of the triangle)

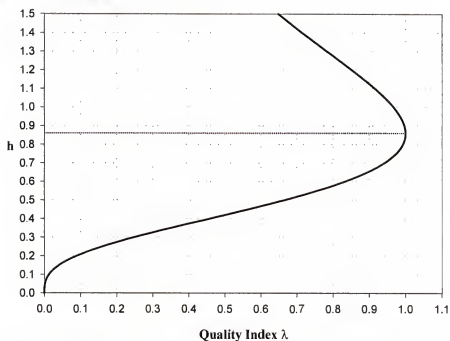


Figure 34 : Variation of quality index along Z-axis (platform with joint separated along one edge of the triangle)

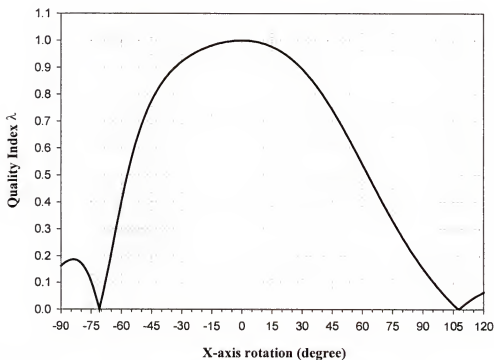


Figure 35 : Variation of quality index when platform rotates about X-axis (platform with joint separated along one edge of the triangle)

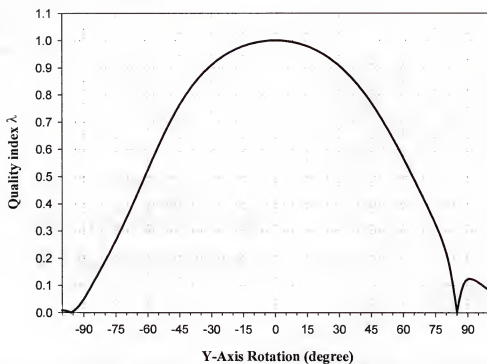


Figure 36 : Variation of quality index when platform rotates about Y-axis (platform with joint separated along one edge of the triangle)

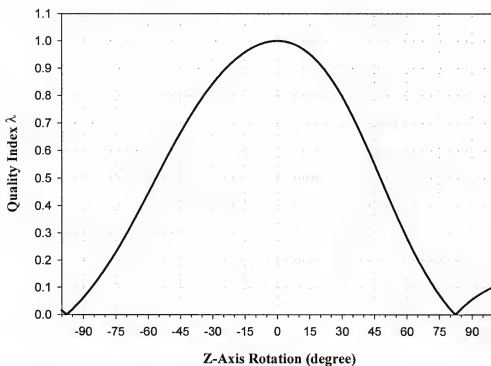


Figure 37 : Variation of quality index when platform rotates about Z-axis (platform with joint separated along one edge of the triangle)

## CHAPTER 9

### DEVELOPMENT OF WEB BASED INTERACTIVE GRAPHICAL DESIGN AND ANALYSIS TOOL

The plots shown earlier present variations of quality indices on x-y-z translations and rotations about x-y-z axes, however, they do not give enough information on how the quality index varies over the platform's workspace. Therefore, an interactive computer graphic simulation program that shows variations of quality indices of a platform device and the corresponding configurations was developed.

Of the many 3D-graphical tools available for this task such as OpenGL, Open Inventor, Direct 3D and VRML, VRML was used for the 3D graphics interface and Java for the programming part. These were chosen because they are standard World Wide Web protocols for 3D graphics and programming language. The simulation is available on the internet and can be run from anywhere in a Java enabled browser that contains a VRML plugin. This can be called "distributed simulation." Distributed simulation has many advantages over conventional programming methods. One of them is data sharing over the internet.

Since it is difficult to find a maximum determinant value of the Jacobian matrix of the platforms, it is necessary to collect all the values of the maximum determinants found previously and store them in a database. This data can then be shared through an internet server.

Java applets are used to provide dynamic content on a Web page and they are also used to deliver data to a client system from the server. An applet can make a connection to the host it was loaded from and it can also send user input data to the server. Java provides several easy-to-use sets of APIs for networking including socket and packet based protocols. However, direct socket connection between client and server is not recommended because of a possible network security problem. Most commercial internet servers do not allow direct socket connection from clients. Even though socket networking is very powerful, it is not going to be used for the development of the design and analysis tool for the platform.

Below is the required hardware and software lists and some brief information about the technologies used to develop the design tool.

### Specifications

#### Hardware

Client : PC running Windows 95/NT or Silicon Graphics Workstation

Server : PC or any workstation running Microsoft Internet Information Server 4.0

#### Software

Client : Web Browser (Netscape Communicator 4.06 or Internet Explorer 4.01)

Cosmo Player 2.1 (1.02 for SGI) plugin

Server : Microsoft Windows NT Workstation or Server 4.0

Microsoft Internet Information Server 4.0

## Programming Languages and tools

### VRML

VRML stands for Virtual Reality Modeling Language. It became a standard 3D graphic language for the Internet around 1996. The current version of VRML is VRML97 which became an International Standard in 1997. In the beginning, VRML was a purely static 3D-scene description language (VRML1.0). The later version 2 (VRML 2.0) introduced animation and programmable behavior. VRML97 is the official standard of VRML 2.0 and 98% compatible with VRML 2.0. One of the new features in VRML97 is the external authoring interface (EAI) with which a VRML scene can be manipulated by an external program. VRML scenes can be prepared by CAD programs such as AutoCAD and Pro/Engineer and then converted to a VRML format.

### Java and JavaScript

Java is a platform-independent programming language and only programming language which VRML can be bound with now. Many Java development tools are available such as JDK, Visual J++ and so on. JDK will be used for compiling Java programs. The version of JDK has changed from 1.0 to 1.1, and soon it will be 1.2. Currently, Java programs are prepared using JDK 1.1, but later, when version 1.2 is released, all the programs will be upgraded.

JavaScript is originally developed by Netscape and its syntax is based on the Java programming language. It is used for creating dynamic content on HTML pages. Since JavaScript is a scripting language it doesn't need to be compiled to be executed.

### VBScript

Microsoft Visual Basic Scripting Edition (VBScript) is a subset of the Microsoft Visual Basic language. It is implemented as an interpreter for use in World Wide Web browsers similar to JavaScript and other applications that use Microsoft ActiveX Controls. VBScript is commonly used for developing Active Server Pages (ASP) which run on the server and not on the client.

### Active Server Page (ASP)

Active Server Pages (ASP) is a server-side scripting environment that creates dynamic Web pages or builds Web applications. ASP pages are files that contain HTML tags, text, and script commands (VBScript). ASP pages can call ActiveX components to perform tasks, such as connecting to a database or performing an external calculation. Microsoft Internet Information Server is the only HTTP server that can process ASP pages.

### Remote Scripting

Remote scripting allows for client/server-type interactions between the browser and the Web server. The web application can validate the user entries without having to reload the page. Specifically, the script must first identify a server page to connect to, then transmit the call to the server. Any return value from the call is transmitted back to the originating script.

### ODBC (Open DataBase Connectivity)

Open Database Connectivity (ODBC) is a widely accepted application programming interface (API) for database access. It is based on the Call-Level Interface (CLI) specifications from X/Open and ISO/IEC for database APIs and uses Structured Query Language (SQL) as its database access language.

When programming to interact with ODBC you only need to talk the ODBC language (a combination of ODBC API function calls and the SQL language). The ODBC Manager will figure out how to contend with the type of database you are targeting. Regardless of the database type used, all of the calls will be to the ODBC API. All that needs is an ODBC driver that is specific to the type of database.

### System Design

Figure-38 illustrates the system structure of the analysis tool. It has client and server side components. The client side components are consist of VRML and Java Applet and the server side components are comprised of ASP and Win32 program which performs a numerical search of the maximum value of the determinant of the Jacobian matrix of the platform. The database used for the project is Microsoft Access and it is provided by ODBC driver installed on the server machine.

Once all the program codes of the analysis tool are loaded to a client machine, A user can evaluate the configuration of the platform using the graphical user interface provided by the Java Applet and the built-in interface of the VRML browser (Cosmo Player). The client machine connects the server again when the user needs to get the maximum value of the determinant of the Jacobian matrix of the platform. The client program sends the configuration data of the platform to the server through the standard



HTTP protocol. The configuration data has two parts, top configuration and the base configuration, and each part consists of x-y coordinates of the six joints coordinates with respect to the center of the top or base platform. Those joint coordinates are divided by the length from the center of the platform to the first joint position. These unitized coordinates are delimited by a character 'x'.

Once the configuration data is received by the ASP page on the server, the ASP page searches the database for the matching configuration. If the match is found, the page sends the corresponding maximum determinant value of the Jacobian matrix back to the client. If there is no matching configuration, the ASP page checks to see if there are any other processes running for the search of the maximum determinant value. If there is one, the ASP page sends a message notifying that there is a running process and asks user to wait until the process ends. Otherwise the page calls a program which performs a numerical search of the maximum determinant of the given configuration. This program updates the database when it finishes the search.

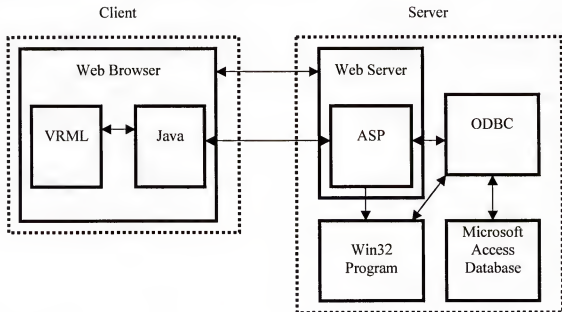


Figure 38 : System Diagram

### Procedures

#### Setup Client Side Components (Integrating Java and VRML)

Java can be used with VRML in two ways. VRML has a Java script node and Java can be used as the scripting language for that node. The Java script interacts with the VRML world through the Java Script Authoring Interface (JSAI). The JSAI allows Java to communicate with other VRML nodes. The Java script receives events from a node, executes the script, and then sends the results to other nodes as events. This allows Java to perform complex modifications to the objects in a VRML scene.

Java can also control a VRML world externally, using an External Authoring Interface (EAI). The EAI allows one or more VRML worlds to be added to a Java program. This allows VRML to be part of a larger or more conventional application. The EAI allows almost the same functionality as the JSAI. Events can be sent to VRML nodes and also node states and characteristics are received from other nodes. However unlike JSAI, a VRML event itself cannot cause EAI program codes to be executed. Rather the Java applet that contains the EAI codes controls a VRML event and nodes. For example, pressing a button under control of the Java applet could cause an event to be sent to the VRML world that would change the characteristics of the scene. The procedure for creating Java/VRML applet programs is as follows:

- (a) Java codes are compiled with either JDK 1.0.2 or JDK 1.1 depending on how the provided browser environment handles the bytecodes (for example, Netscape Navigator 4.0 does not fully handle JDK 1.1 produced bytecode).
- (b) The EAI classes are shipped with an EAI enabled VRML browser and they are not part of the standard JDK classes. Therefore the CLASSPATH environment

variable must be set to point to the directory in which they are placed. EAI classes are included in the “vrmI.external.\*” packages. These packages contain all the classes and methods needed to access the VRML world.

- (c) Before accessing the VRML world, a reference to the Browser class instance must be set. This class is the reflection of the VRML world in the Java applet, and it is used for all access to the world. The static `Browser.getBrowser( )` method is used to get the reference.
- (d) Communicating with a particular node is done by first getting a reference to the node, using the `getNode( )` method. Because only named nodes can be accessed from the EAI, a string corresponding to the DEF name of a node in the VRML scene is passed as a method argument.
- (e) All VRML EAI initializations are set up in the `start()` method, and not in `init()`. This avoids the problem of stale Browser and Node pointers when the user leaves and then revisits the web page containing this applet.
- (f) Once you have a reference to a node you can send events to its `eventIns` using `getEventIn( )` method and get the current value of its `eventOuts` using the `getEventOut()` method.
- (g) The `createVrmIFromString( )` method is used to create nodes from scratch and add them to the scene.

### Setup Server Side Components

A database is set up using Microsoft Access on the server and will be used to hold all the maximum values of the determinant of the platforms found by client or calculated by server. The database table and fields are listed in Figure-39. Once the database file is

created it must be connected to the ODBC to be accessed from Active Server Pages using the ODBC menu of Control Panel on the server machine.

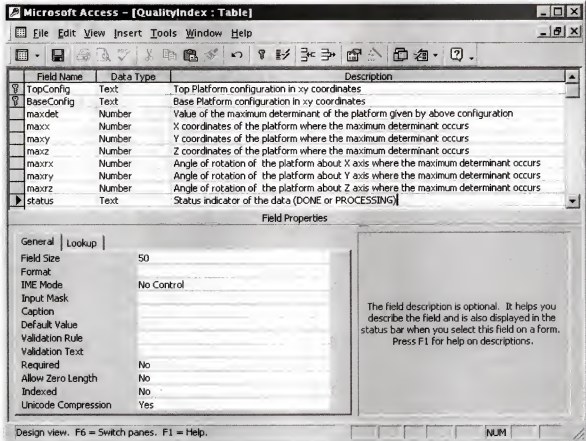


Figure 39 : Database table information

An ASP page is prepared for handling the database. The Java applet which controls the client VRML scene connects to this ASP page through a HTTP server to get the maximum determinant values from the database. When the ASP page receives the configuration data from the client, it tries to find the matching configuration from the database. If it finds one, it returns the corresponding maximum determinant value. If it fails to find the matching configuration, it searches for 'PROCESSING' in 'status' field of the database. If it finds 'PROCESSING', the ASP page sends 'WAIT' message back to the client and if there is no entry of 'PROCESSING' the ASP does as follows:

- a) It adds a data entry of the received configuration to the database with 'PROCESSING' in the 'status' field
- b) It executes the search program.
- c) It sends a 'PROCESSING' message back to the client.

The program which performs a numerical search of the maximum determinant value of the Jacobian matrix is coded using C++ programming language for the fast calculation. This program has ODBC APIs used for connecting the database, getting the configuration data and updating the database entry after finishing the search.

### Results

Figure-40 shows the client side view of the design tool. The left side of the browser screen displays the three-dimensional graphics of the platform and the right side displays the controls of the platform. A user can set up the configuration of the platform, control the position and orientation of the platform, and view the Jacobian matrix and the determinant value with the help of the buttons on the main control panel.

Clicking the "Configuration" button brings up a dialog box shown in Figure-41. Using this dialog, a user can modify the geometry of the platform. There are two sections controlling either top or base. The menu "Type" indicates geometry of the platform and "Size" indicates size of the platform -  $a$  or  $b$  discussed earlier in this paper. If the user selects "Hexagon", the joint separation menu will be available for modifications. The amount of the separation is shown as a percentage value, 0 to 100 instead of 0 to 1.

The two read colored ball joint on the platform indicates the connector leg 1 and the joint numbers of the rest of the connector legs are assigned in counter clockwise.

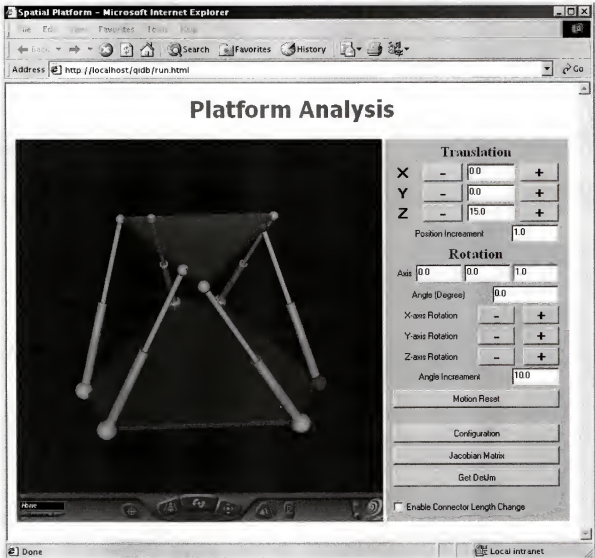


Figure 40 : Web based interactive analysis tool for platform manipulators

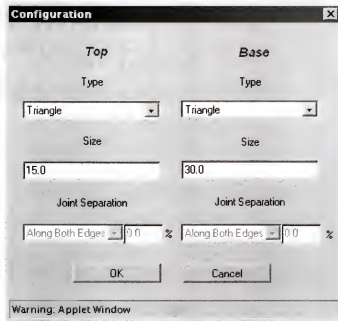


Figure 41 : Platform configuration dialog

Pushing the button “Jacobian Matrix” will open up an another dialog (see Figure-42) displaying the Jacobian matrix, its determinant value and the quality index if the maximum determinant value of the platform is available.

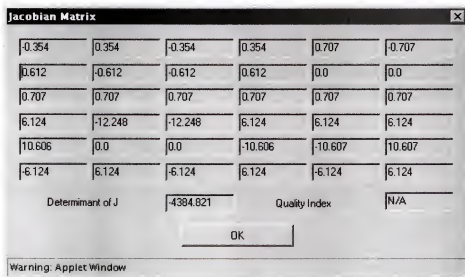


Figure 42 : Dialog displaying Jacobian Matrix

## CHAPTER 10 CONCLUSION

A quality index, see equation (1), can be easily used for every fully in-parallel six-linear-actuator manipulator at any configuration once the maximum values of the determinant of the Jacobian matrix for the platforms are known. Its maximum value,  $\lambda = 1$ , is attached to just one particular central configuration of one device and the dimensions are specified.

This study supports the conclusion of Hunt and McAree (1998) that, departure from an octahedral in the direction of the general 6-6 form progressively leads to poorer performance and unworkability coupled with considerably greater analytical and algorithmic complications.

Some reduction of quality index  $\lambda$  below 1 is, of course, inevitable as the manipulator is displaced; it is up to both the designer and the user to make an intelligent judgement of a lower limit for some specific application.

No startling surprises are here revealed; in fact these results and examples are in remarkable conformity with common sense. A more complicated formulation of the quality index, such as one that includes a penalty for the shear bulk of the manipulator has been deliberately avoided. The imprecision inherent in such an elaboration detracts from the simplicity of this quality index  $\lambda$ .

On the matter of avoidable actuator-range, it is recognized that a designer may need to exercise considerable ingenuity in achieving greater leg-elongation without both reduction of structural integrity and excessive degradation of  $\lambda$ .



It is usually difficult to get the maximum value of the platform with non-symmetrical joint connections. The practical solution of the difficulties is provided by developing a tool that will enable the sharing of the design information over the internet using the latest technology.

In this paper, only the non-redundant six-degree-of-freedom platform devices are analyzed. A certain singular configuration can be easily avoided by adding more legs to the platform which makes the platform redundant. It would be fruitful to expand the research to the redundant platforms.

## APPENDIX PROGRAM LISTS

### 1. CONFIG\_DATA.JAVA

```
//Title:      VRML Platform
//Copyright:   Copyright (c) 2000
//Author:      Jaehoon Lee

package Platform;

/*****
/* platform configuration data class */
*****/

public class Config_data
{
    boolean maxdetfound = true;
    float maxdet = 4384.2537f;
    float[] maxpos = new float[3];
    float[] maxrot = new float[3];
    float top_size = 15.0f;
    float base_size = 30.0f;
    float alpha = 10.0f;
    float beta = 10.0f;
    String top_shape = new String("Triangle");
    String base_shape = new String("Triangle");
    String top_separation = new String("Along Both Edges");
    String base_separation = new String("Along Both Edges");

    // constructor
    public Config_data()
    {
        maxpos[0]=0.0f; maxpos[1]=0.0f; maxpos[2]=15.0f;
        maxrot[0]=0.0f; maxrot[1]=0.0f; maxrot[2]=0.0f;
    }

    // methods
    public float getTopSize() {return top_size;}
    public float getBaseSize() {return base_size;}
    public float getAlpha() {return alpha;}
    public float getBeta() {return beta;}
    public String getTopShape() {return top_shape;}
    public String getBaseShape() {return base_shape;}
    public String getTopSeparation() {return top_separation;}
    public String getBaseSeparation() {return base_separation;}
    public float getMaxdet() {return maxdet;}
    public float[] getMaxpos() {return maxpos;}
    public float[] getMaxrot() {return maxrot;}
    public boolean getMaxdetFound() {return maxdetfound;}

    public void setTopSize(float f) {top_size=f;}
    public void setBaseSize(float f) {base_size=f;}
    public void setAlpha(float f) {alpha=f;}
    public void setBeta(float f) {beta=f;}
    public void setTopShape(String s) {top_shape=s;}
    public void setBaseShape(String s) {base_shape=s;}
    public void setTopSeparation(String s) {top_separation=s;}
    public void setBaseSeparation(String s) {base_separation=s;}
    public void setMaxdet(float mj) {maxdet = mj;}
    public void setMaxdetFound(boolean truefalse) {maxdetfound=truefalse;}
```

```

public void setMaxpos(float[] pos) {
    maxpos[0]=pos[0];  maxpos[1]=pos[1];  maxpos[2]=pos[2];
}

public void setMaxpos(float x, float y, float z) {
    maxpos[0]=x;  maxpos[1]=y;  maxpos[2]=z;
}

public void setMaxrot(float[] rot) {
    maxrot[0]=rot[0];  maxrot[1]=rot[1];  maxrot[2]=rot[2];
}

public void setMaxrot(float rotx,float roty,float rotz) {
    maxrot[0]=rotx;  maxrot[1]=roty;  maxrot[2]=rotz;
}
}

```

## 2. CONFIG\_DIALOG.JAVA

```

//Title:      VRML Platform
//Copyright:   Copyright (c) 2000
//Author:      Jaehoon Lee

package Platform;

import java.awt.*;
import java.awt.event.*;
import com.borland.jbcl.layout.*;
import vrml.external.field.*;
import vrml.external.exception.*;
import vrml.external.Node;
import vrml.external.Browser;

/*****
/* configuration dialog class
*****/

public class Config_Dialog extends Dialog
{
    Panel panell = new Panel();
    XYLayout xYLayout1 = new XYLayout();
    Choice top_type = new Choice();
    Choice base_type = new Choice();
    Choice top_separation = new Choice();
    Choice base_separation = new Choice();
    Label label_Top = new Label();
    Label label_Base = new Label();
    Label label_top_sep = new Label();
    Label label_base_sep = new Label();
    Label label_alpha_per = new Label();
    Label label_beta_per = new Label();
    Label label_top_type = new Label();
    Label label_base_type = new Label();
    Label label_top_size = new Label();
    Label label_base_size = new Label();
    TextField alpha = new TextField();
    TextField beta = new TextField();
    TextField top_size = new TextField();
    TextField base_size = new TextField();
    Button button_ok = new Button();
    Button button_cancel = new Button();

    // platform configuration data
    Config_data pdata = null;

    // variables connecting VRML scene
    public EventInSFFloat stt = null;
    public EventInSFFloat stb = null;
}

```

```

public EventInSFVec3f sht = null;
public EventInSFVec3f shb = null;
public EventInSFBool sct = null;
public EventInSFBool scb = null;

// constructor
public Config_Dialog(Frame frame, String title, boolean modal, Config_data pd)
{
    super(frame, title, modal);
    enableEvents(AWTEvent.WINDOW_EVENT_MASK);
    pdata = pd;
    try
    {
        jbInit();
        add(panell);
        pack();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
}

void jbInit() throws Exception
{
    panell.setLayout(xYLayoutl);

    xYLayoutl.setHeight(300);
    xYLayoutl.setWidth(360);

    label_Top.setFont(new java.awt.Font("SansSerif", 3, 14));
    label_Top.setForeground(Color.blue);
    label_Top.setAlignment(1);
    label_Top.setText("Top");

    label_Base.setFont(new java.awt.Font("SansSerif", 3, 14));
    label_Base.setForeground(Color.blue);
    label_Base.setAlignment(1);
    label_Base.setText("Base");

    label_top_sep.setAlignment(1);
    label_top_sep.setText("Joint Separation");
    label_base_sep.setAlignment(1);
    label_base_sep.setText("Joint Separation");
    label_alpha_per.setText("%");
    label_beta_per.setText("%");
    label_top_type.setAlignment(1);
    label_top_type.setText("Type");
    label_base_type.setAlignment(1);
    label_base_type.setText("Type");
    label_top_size.setAlignment(1);
    label_top_size.setText("Size");
    label_base_size.setAlignment(1);
    label_base_size.setText("Size");

    top_type.addItem("Triangle");
    top_type.addItem("Hexagon");
    top_type.addItem("Custom");
    top_type.select(pdata.getTopShape());
    top_type.addItemListener(new java.awt.event.ItemListener()
    {

        public void itemStateChanged(ItemEvent e)
        {
            top_type_itemStateChanged(e);
        }
    });

    base_type.addItem("Triangle");
    base_type.addItem("Hexagon");
    base_type.addItem("Custom");
}

```

```

base_type.select(pdata.getBaseShape());
base_type.addItemListener(new java.awt.event.ItemListener()
{
    public void itemStateChanged(ItemEvent e)
    {
        base_type_itemStateChanged(e);
    }
});

if (pdata.getTopShape().equals("Hexagon"))
{
    top_size.setEnabled(true);
    top_separation.setEnabled(true);
    alpha.setEnabled(true);
}
else
{
    if (pdata.getTopShape().equals("Custom"))
        top_size.setEnabled(false);
    else
        top_size.setEnabled(true);
    top_separation.setEnabled(false);
    alpha.setEnabled(false);
}

if (pdata.getBaseShape().equals("Hexagon"))
{
    base_size.setEnabled(true);
    base_separation.setEnabled(true);
    beta.setEnabled(true);
}
else
{
    if (pdata.getBaseShape().equals("Custom"))
        base_size.setEnabled(false);
    else
        base_size.setEnabled(true);
    base_separation.setEnabled(false);
    beta.setEnabled(false);
}

top_size.setText(Float.toString(pdata.getTopSize()));
base_size.setText(Float.toString(pdata.getBaseSize()));

top_separation.addItem("Along Both Edges");
top_separation.addItem("Along One Edge");
top_separation.addItem("Toward Center");
top_separation.select(pdata.getTopSeparation());

base_separation.addItem("Along Both Edges");
base_separation.addItem("Along One Edge");
base_separation.addItem("Toward Center");
base_separation.select(pdata.getBaseSeparation());

alpha.setText(Float.toString(pdata.getAlpha()));
beta.setText(Float.toString(pdata.getBeta()));

button_ok.setLabel("OK");
button_ok.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        button_ok_actionPerformed(e);
    }
});

button_cancel.setLabel("Cancel");
button_cancel.addActionListener(new java.awt.event.ActionListener()
{

```

```

public void actionPerformed(ActionEvent e)
{
    button_cancel_actionPerformed(e);
}

});

panell.add(label_Top, new XYConstraints(14, 17, 161, 26));
panell.add(label_Base, new XYConstraints(186, 18, 161, 26));
panell.add(label_alpha_per, new XYConstraints(167, 217, 17, -1));
panell.add(label_top_type, new XYConstraints(15, 52, 150, 22));
panell.add(label_base_type, new XYConstraints(187, 52, 150, 22));
panell.add(label_base_size, new XYConstraints(187, 116, 150, 23));
panell.add(top_type, new XYConstraints(15, 84, 150, 23));
panell.add(base_type, new XYConstraints(187, 83, 150, 23));
panell.add(top_separation, new XYConstraints(15, 217, 110, 23));
panell.add(base_separation, new XYConstraints(187, 217, 110, 23));
panell.add(alpha, new XYConstraints(124, 217, 44, 23));
panell.add(beta, new XYConstraints(296, 217, 44, 23));
panell.add(top_size, new XYConstraints(15, 149, 150, -1));
panell.add(base_size, new XYConstraints(187, 148, 150, -1));
panell.add(label_top_sep, new XYConstraints(15, 181, 150, 26));
panell.add(label_base_sep, new XYConstraints(187, 180, 150, 26));
panell.add(label_top_size, new XYConstraints(15, 116, 150, -1));
panell.add(button_ok, new XYConstraints(69, 260, 96, -1));
panell.add(button_cancel, new XYConstraints(189, 260, 96, -1));
panell.add(label_beta_per, new XYConstraints(342, 218, 17, -1));

}

void cancel()
{
    dispose();
}

protected void processWindowEvent(WindowEvent e)
{
    if (e.getID() == WindowEvent.WINDOW_CLOSING)
    {
        cancel();
    }
    super.processWindowEvent(e);
}

/*****
/* event handlers for the AWT components */
/* (button, dropdown menus, text boxes, etc.) */
*****/

void top_type_itemStateChanged(ItemEvent e)
{
    if (e.getItem().equals("Hexagon"))
    {
        top_size.setEnabled(true);
        top_separation.setEnabled(true);
        alpha.setEnabled(true);
    }
    else
    {
        if (e.getItem().equals("Custom"))
        {
            top_size.setEnabled(false);
        }
        else
        {
            top_size.setEnabled(true);
            top_separation.setEnabled(false);
            alpha.setEnabled(false);
        }
    }
}

void base_type_itemStateChanged(ItemEvent e)
{

```

```

if (e.getItem().equals("Hexagon"))
{
    base_size.setEnabled(true);
    base_separation.setEnabled(true);
    beta.setEnabled(true);
}
else
{
    if (e.getItem().equals("Custom"))
        base_size.setEnabled(false);
    else
        base_size.setEnabled(true);
    base_separation.setEnabled(false);
    beta.setEnabled(false);
}
}

void button_ok_actionPerformed(ActionEvent e)
{
    float[] jdata = new float[3];
    float djm,h, a,b, ap,bt;

    // set the config data according to the user input
    pdata.setTopShape(top_type.getSelectedItem());
    pdata.setBaseShape(base_type.getSelectedItem());

    pdata.setTopSeparation(top_separation.getSelectedItem());
    pdata.setBaseSeparation(base_separation.getSelectedItem());

    a=Float.valueOf(top_size.getText()).floatValue();
    b=Float.valueOf(base_size.getText()).floatValue();
    pdata.setTopSize(a);
    pdata.setBaseSize(b);

    pdata.setAlpha(Float.valueOf(alpha.getText()).floatValue());
    pdata.setBeta(Float.valueOf(beta.getText()).floatValue());

    ap=Float.valueOf(alpha.getText()).floatValue()/100.0f;
    bt=Float.valueOf(beta.getText()).floatValue()/100.0f;

    if (pdata.getTopShape().equals("Triangle"))
    {
        top_separation.setEnabled(false);
        alpha.setEnabled(false);
        stt.setValue(pdata.getTopSize());
    }
    if (pdata.getBaseShape().equals("Triangle"))
    {
        base_separation.setEnabled(false);
        beta.setEnabled(false);
        stb.setValue(pdata.getBaseSize());
    }
    if (pdata.getTopShape().equals("Custom"))
    {
        top_separation.setEnabled(false);
        alpha.setEnabled(false);
        sct.setValue(true);
    }
    if (pdata.getBaseShape().equals("Custom"))
    {
        base_separation.setEnabled(false);
        beta.setEnabled(false);
        scb.setValue(true);
    }
    if (pdata.getTopShape().equals("Hexagon"))
    {
        jdata[0] = pdata.getTopSize();
        jdata[1] = pdata.getAlpha();
        if (pdata.getTopSeparation().equals("Along Both Edges"))

```

```

    jdata[2]=0.0f;
}
if (pdata.getTopSeparation().equals("Along One Edge"))
{
    jdata[2]=1.0f;
}
if (pdata.getTopSeparation().equals("Toward Center"))
{
    jdata[2]=2.0f;
}
sht.setValue(jdata);
}

if (pdata.getBaseShape().equals("Hexagon"))
{
    jdata[0] = pdata.getBaseSize();
    jdata[1] = pdata.getBeta();
    if (pdata.getBaseSeparation().equals("Along Both Edges"))
    {
        jdata[2]=0.0f;
    }
    if (pdata.getBaseSeparation().equals("Along One Edge"))
    {
        jdata[2]=1.0f;
    }
    if (pdata.getBaseSeparation().equals("Toward Center"))
    {
        jdata[2]=2.0f;
    }
    shb.setValue(jdata);
}

h=a;
djm=-0.0001f;
pdata.setMaxdetFound(false);

// calculate known DetJm
if (pdata.getTopShape().equals("Triangle"))
{
    if (pdata.getBaseShape().equals("Triangle"))
    {
        djm=(float) (27.0*a*a*b*b*b/(32*Math.sqrt(Math.pow(a*a-a*b+b*b,3))));
        h=(float) (Math.sqrt((a*a-a*b+b*b)/3));
        pdata.setMaxdetFound(true);
    }
    if (pdata.getBaseShape().equals("Hexagon"))
        && pdata.getBaseSeparation().equals("Along Both Edges"))
    {
        djm=(float) (27.0*a*a*b*b*b*Math.pow(2*bt-1,3)
            /(32*Math.sqrt(Math.pow(a*a-a*b+b*b*(3*bt*bt-3*bt+1),3))));
        h=(float) (Math.sqrt((a*a-a*b+b*b*(3*bt*bt-3*bt+1)/3)));
        pdata.setMaxdetFound(true);
    }
}

if (pdata.getTopShape().equals("Hexagon"))
    && pdata.getBaseShape().equals("Hexagon")
    && pdata.getTopSeparation().equals("Along Both Edges")
    && pdata.getBaseSeparation().equals("Along Both Edges"))
{
    djm=(float) (27.0*a*a*b*b*b*Math.pow(3*ap*bt-2*ap-2*bt+1,3)
        /(32*Math.sqrt(Math.pow(a*a*(3*ap*ap-3*ap+1)
            +a*b*(3*ap*bt-1)+b*b*(3*bt*bt-3*bt+1),3))));
    h=(float) (Math.sqrt((a*a*(3*ap*ap-3*ap+1)
        +a*b*(3*ap*bt-1)+b*b*(3*bt*bt-3*bt+1)/3)));
    pdata.setMaxdetFound(true);
}

pdata.setMaxdet(Math.abs(djm));
pdata.setMaxpos(0.0f,0.0f,h);

```



```

        pdata.setMaxrot(0.0f,0.0f,0.0f);
    }
    dispose();
}

void button_cancel_actionPerformed(ActionEvent e)
{
    dispose();
}
}

```

### 3. JB\_DIALOG.JAVA

```

//Title:          VRML Platform
//Copyright:      Copyright (c) 2000
//Author:         Jaehoon Lee

package Platform;

import java.awt.*;
import java.awt.event.*;
import com.borland.jbcl.layout.*;

/*****
/* Jacobian matrix, DetJ and Quality index display dialog class */
*****/

public class JB_Dialog extends Dialog
{
    Panel panell = new Panel();
    XYLayout xYLayout1 = new XYLayout();
    TextField textFieldd11 = new TextField();
    TextField textFieldd12 = new TextField();
    TextField textFieldd13 = new TextField();
    TextField textFieldd14 = new TextField();
    TextField textFieldd15 = new TextField();
    TextField textFieldd16 = new TextField();
    TextField textFieldd21 = new TextField();
    TextField textFieldd22 = new TextField();
    TextField textFieldd23 = new TextField();
    TextField textFieldd24 = new TextField();
    TextField textFieldd25 = new TextField();
    TextField textFieldd26 = new TextField();
    TextField textFieldd31 = new TextField();
    TextField textFieldd32 = new TextField();
    TextField textFieldd33 = new TextField();
    TextField textFieldd34 = new TextField();
    TextField textFieldd35 = new TextField();
    TextField textFieldd36 = new TextField();
    TextField textFieldd41 = new TextField();
    TextField textFieldd42 = new TextField();
    TextField textFieldd43 = new TextField();
    TextField textFieldd44 = new TextField();
    TextField textFieldd45 = new TextField();
    TextField textFieldd46 = new TextField();
    TextField textFieldd51 = new TextField();
    TextField textFieldd52 = new TextField();
    TextField textFieldd53 = new TextField();
    TextField textFieldd54 = new TextField();
    TextField textFieldd55 = new TextField();
    TextField textFieldd56 = new TextField();
    TextField textFieldd61 = new TextField();
    TextField textFieldd62 = new TextField();
    TextField textFieldd63 = new TextField();
    TextField textFieldd64 = new TextField();
    TextField textFieldd65 = new TextField();
}

```

```

TextField textField66 = new TextField();
Label label1 = new Label();
TextField textField_det = new TextField();
TextField textField_qi = new TextField();
Label label2 = new Label();
Button ok_button = new Button("Close");

// constructor
public JB_Dialog(Frame frame, String title, boolean modal, float[][] jmat,
                 float detj, float qi)
{
    super(frame, title, modal);
    enableEvents(AWTEvent.WINDOW_EVENT_MASK);
    try
    {
        jbInit();
        add(panel1);
        pack();
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }

    textField11.setText(Float.toString((float)Math.round(jmat[0][0]*1000)/1000f));
    textField12.setText(Float.toString((float)Math.round(jmat[0][1]*1000)/1000f));
    textField13.setText(Float.toString((float)Math.round(jmat[0][2]*1000)/1000f));
    textField14.setText(Float.toString((float)Math.round(jmat[0][3]*1000)/1000f));
    textField15.setText(Float.toString((float)Math.round(jmat[0][4]*1000)/1000f));
    textField16.setText(Float.toString((float)Math.round(jmat[0][5]*1000)/1000f));
    textField21.setText(Float.toString((float)Math.round(jmat[1][0]*1000)/1000f));
    textField22.setText(Float.toString((float)Math.round(jmat[1][1]*1000)/1000f));
    textField23.setText(Float.toString((float)Math.round(jmat[1][2]*1000)/1000f));
    textField24.setText(Float.toString((float)Math.round(jmat[1][3]*1000)/1000f));
    textField25.setText(Float.toString((float)Math.round(jmat[1][4]*1000)/1000f));
    textField26.setText(Float.toString((float)Math.round(jmat[1][5]*1000)/1000f));
    textField31.setText(Float.toString((float)Math.round(jmat[2][0]*1000)/1000f));
    textField32.setText(Float.toString((float)Math.round(jmat[2][1]*1000)/1000f));
    textField33.setText(Float.toString((float)Math.round(jmat[2][2]*1000)/1000f));
    textField34.setText(Float.toString((float)Math.round(jmat[2][3]*1000)/1000f));
    textField35.setText(Float.toString((float)Math.round(jmat[2][4]*1000)/1000f));
    textField36.setText(Float.toString((float)Math.round(jmat[2][5]*1000)/1000f));
    textField41.setText(Float.toString((float)Math.round(jmat[3][0]*1000)/1000f));
    textField42.setText(Float.toString((float)Math.round(jmat[3][1]*1000)/1000f));
    textField43.setText(Float.toString((float)Math.round(jmat[3][2]*1000)/1000f));
    textField44.setText(Float.toString((float)Math.round(jmat[3][3]*1000)/1000f));
    textField45.setText(Float.toString((float)Math.round(jmat[3][4]*1000)/1000f));
    textField46.setText(Float.toString((float)Math.round(jmat[3][5]*1000)/1000f));
    textField51.setText(Float.toString((float)Math.round(jmat[4][0]*1000)/1000f));
    textField52.setText(Float.toString((float)Math.round(jmat[4][1]*1000)/1000f));
    textField53.setText(Float.toString((float)Math.round(jmat[4][2]*1000)/1000f));
    textField54.setText(Float.toString((float)Math.round(jmat[4][3]*1000)/1000f));
    textField55.setText(Float.toString((float)Math.round(jmat[4][4]*1000)/1000f));
    textField56.setText(Float.toString((float)Math.round(jmat[4][5]*1000)/1000f));
    textField61.setText(Float.toString((float)Math.round(jmat[5][0]*1000)/1000f));
    textField62.setText(Float.toString((float)Math.round(jmat[5][1]*1000)/1000f));
    textField63.setText(Float.toString((float)Math.round(jmat[5][2]*1000)/1000f));
    textField64.setText(Float.toString((float)Math.round(jmat[5][3]*1000)/1000f));
    textField65.setText(Float.toString((float)Math.round(jmat[5][4]*1000)/1000f));
    textField66.setText(Float.toString((float)Math.round(jmat[5][5]*1000)/1000f));
    textField_det.setText(Float.toString((float)Math.round(detj*1000)/1000f));
    //textField_det.setText("N/A");
    if (qi >= 0.0f)
    textField_qi.setText(Float.toString((float)Math.round(qi*1000)/1000f));
    else textField_qi.setText("N/A");
}

void jbInit() throws Exception
{
    panel1.setLayout(xYLayout1);
    xYLayout1.setHeight(258);
}

```

```

xYLayout1.setWidth(509);
textField11.setEditable(false);
textField12.setEditable(false);
textField13.setEditable(false);
textField14.setEditable(false);
textField15.setEditable(false);
textField16.setEditable(false);
textField21.setEditable(false);
textField22.setEditable(false);
textField23.setEditable(false);
textField24.setEditable(false);
textField25.setEditable(false);
textField26.setEditable(false);
textField31.setEditable(false);
textField32.setEditable(false);
textField33.setEditable(false);
textField34.setEditable(false);
textField35.setEditable(false);
textField36.setEditable(false);
textField41.setEditable(false);
textField42.setEditable(false);
textField43.setEditable(false);
textField44.setEditable(false);
textField45.setEditable(false);
textField46.setEditable(false);
textField51.setEditable(false);
textField52.setEditable(false);
textField53.setEditable(false);
textField54.setEditable(false);
textField55.setEditable(false);
textField56.setEditable(false);
textField61.setEditable(false);
textField62.setEditable(false);
textField63.setEditable(false);
textField64.setEditable(false);
textField65.setEditable(false);
textField66.setEditable(false);
label1.setAlignment(1);
label1.setText("Determinant of J");
textField_det.setEditable(false);
textField_qi.setEditable(false);
label2.setText("Quality Index");
label2.setAlignment(1);
ok_button.setName("OK");
ok_button.setLabel("OK");
ok_button.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ok_button_actionPerformed(e);
    }
});
panel1.add(textField21, new XYConstraints(13, 42, 77, -1));
panel1.add(textField22, new XYConstraints(94, 42, 77, -1));
panel1.add(textField23, new XYConstraints(175, 42, 77, -1));
panel1.add(textField24, new XYConstraints(256, 42, 77, -1));
panel1.add(textField25, new XYConstraints(337, 42, 77, -1));
panel1.add(textField26, new XYConstraints(418, 42, 77, -1));
panel1.add(textField31, new XYConstraints(13, 69, 77, -1));
panel1.add(textField32, new XYConstraints(94, 69, 77, -1));
panel1.add(textField33, new XYConstraints(175, 69, 77, -1));
panel1.add(textField34, new XYConstraints(256, 69, 77, -1));
panel1.add(textField35, new XYConstraints(337, 69, 77, -1));
panel1.add(textField36, new XYConstraints(418, 69, 77, -1));
panel1.add(textField41, new XYConstraints(13, 97, 77, -1));
panel1.add(textField42, new XYConstraints(94, 97, 77, -1));
panel1.add(textField43, new XYConstraints(175, 97, 77, -1));
panel1.add(textField44, new XYConstraints(256, 97, 77, -1));
panel1.add(textField45, new XYConstraints(337, 97, 77, -1));
panel1.add(textField46, new XYConstraints(418, 97, 77, -1));

```

```

panell.add(textField51, new XYConstraints(13, 124, 77, -1));
panell.add(textField52, new XYConstraints(94, 124, 77, -1));
panell.add(textField53, new XYConstraints(175, 124, 77, -1));
panell.add(textField54, new XYConstraints(256, 124, 77, -1));
panell.add(textField55, new XYConstraints(337, 124, 77, -1));
panell.add(textField56, new XYConstraints(418, 124, 77, -1));
panell.add(textField61, new XYConstraints(13, 152, 77, -1));
panell.add(textField62, new XYConstraints(94, 152, 77, -1));
panell.add(textField63, new XYConstraints(175, 152, 77, -1));
panell.add(textField64, new XYConstraints(256, 152, 77, -1));
panell.add(textField65, new XYConstraints(337, 152, 77, -1));
panell.add(textField66, new XYConstraints(418, 152, 77, -1));
panell.add(textField16, new XYConstraints(418, 14, 77, -1));
panell.add(textField11, new XYConstraints(13, 14, 77, -1));
panell.add(textField12, new XYConstraints(94, 14, 77, -1));
panell.add(textField13, new XYConstraints(175, 14, 77, -1));
panell.add(textField14, new XYConstraints(256, 14, 77, -1));
panell.add(textField15, new XYConstraints(337, 14, 77, -1));
panell.add(label2, new XYConstraints(256, 184, 144, 25));
panell.add(textField_qi, new XYConstraints(418, 184, 77, 22));
panell.add(textField_det, new XYConstraints(175, 186, 77, 22));
panell.add(label1, new XYConstraints(13, 183, 142, 25));
panell.add(ok_button, new XYConstraints(193, 221, 124, 24));
}

protected void processWindowEvent(WindowEvent e)
{
    if(e.getID() == WindowEvent.WINDOW_CLOSING)
    {
        cancel();
    }
    super.processWindowEvent(e);
}

void cancel()
{
    dispose();
}

void ok_button_actionPerformed(ActionEvent e)
{
    dispose();
}
}

```

#### 4. PLATFORM.JAVA

```

//Title:      VRML Platform
//Copyright:  Copyright (c) 2000
//Author:     Jaehoon Lee

package Platform;

import java.net.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import vrml.external.field.*;
import vrml.external.exception.*;
import vrml.external.Node;
import vrml.external.Browser;
import com.borland.jbcl.layout.*;

/*****
 * Main class for the platform project
 */

```

```

/*****
public class Platform extends Applet
{
    Node xfplatform = null;
    Node xfscrip = null;

    Node[] legnodes = new Node[12];

    EventInSFRotation rot = null;
    EventInSFRotation iprot = null;
    EventOutSFRotation oprot = null;

    EventInSFVec3f trn = null;
    EventInSFVec3f rst = null;

    EventInSFVec3f stt = null;
    EventInSFVec3f stb = null;

    EventInSFVec3f sht = null;
    EventInSFVec3f shb = null;

    EventInSFVec3f sct = null;
    EventInSFVec3f scb = null;

    EventInSFVec3f eclc = null;

    float p_incm = 1.0f;
    float a_incm = 10.0f;

    Browser vrmlbrowser = null;

    XYLayout xYLayout1 = new XYLayout();
    boolean isStandalone = false;
    Button txminus = new Button();
    Button txplus = new Button();
    Button tyminus = new Button();
    Button typlus = new Button();
    Button tzminus = new Button();
    Button tzplus = new Button();
    Button rxplus = new Button();
    Button rxminus = new Button();
    Button ryminus = new Button();
    Button ryplus = new Button();
    Button rzminus = new Button();
    Button rzplus = new Button();
    Button configuration = new Button();
    TextField ty = new TextField();
    TextField tx = new TextField();
    TextField tz = new TextField();
    TextField p_inc = new TextField();
    TextField rxz = new TextField();
    TextField rxy = new TextField();
    TextField rxx = new TextField();
    TextField rxa = new TextField();
    TextField a_inc = new TextField();
    Label label1 = new Label();
    Label label2 = new Label();
    Label label3 = new Label();
    Label label6 = new Label();
    Label label7 = new Label();
    Label label8 = new Label();
    Label label9 = new Label();
    Label label10 = new Label();
    Label label11 = new Label();
    Label label14 = new Label();
    Button button15 = new Button();
    Label label15 = new Label();
    Label label12 = new Label();
    Button jacobian = new Button();
    Button reserved = new Button();

```

```

Button motion_reset = new Button();

Config_data pdata = new Config_data();
Checkbox checkbox1 = new Checkbox();

JB_Dialog Djacobian = null;
Config_Dialog config = null;

//Construct the applet
public Platform()
{
    float[] initpos = {0.0f, 0.0f, 15.0f};
    pdata.setTopShape("Triangle");
    pdata.setBaseShape("Triangle");
    pdata.setTopSeparation("Along Both Edges");
    pdata.setBaseSeparation("Along Both Edges");
    pdata.setTopSize(15.0f);
    pdata.setBaseSize(30.0f);
    pdata.setAlpha(0.0f);
    pdata.setBeta(0.0f);
    pdata.setMaxpos(initpos);
}

//Initialize the applet

public void init()
{
    try
    {
        jbInit();
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }

    tx.setText(Float.toString(pdata.getMaxpos()[0]));
    ty.setText(Float.toString(pdata.getMaxpos()[1]));
    tz.setText(Float.toString(pdata.getMaxpos()[2]));
    rxx.setText(Float.toString(0.0f));
    rxy.setText(Float.toString(0.0f));
    rxz.setText(Float.toString(1.0f));
    rxa.setText(Float.toString(0.0f));
    p_inc.setText(Float.toString(p_incm));
    a_inc.setText(Float.toString(a_incm));
}

private void jbInit() throws Exception
{
    xYLayout1.setWidth(250);
    xYLayout1.setHeight(520);
    txminus.setFont(new Font("Dialog", 1, 20));
    txminus.setLabel("-");
    txminus.addActionListener(new java.awt.event.ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            txminus_actionPerformed(e);
        }
    });
    txplus.setFont(new Font("Dialog", 1, 20));
    txplus.setLabel("+");
    txplus.addActionListener(new java.awt.event.ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            txplus_actionPerformed(e);
        }
    });
}

```

```

tyminus.setFont(new Font("Dialog", 1, 20));
tyminus.setLabel("-");
tyminus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        tyminus_actionPerformed(e);
    }
});
typlus.setFont(new Font("Dialog", 1, 20));
typlus.setLabel("+");
typlus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        typlus_actionPerformed(e);
    }
});
tzminus.setFont(new Font("Dialog", 1, 20));
tzminus.setLabel("-");
tzminus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        tzminus_actionPerformed(e);
    }
});
tzplus.setFont(new Font("Dialog", 1, 20));
tzplus.setLabel("+");
tzplus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        tzplus_actionPerformed(e);
    }
});
rxplus.setFont(new Font("Dialog", 1, 20));
rxplus.setLabel("+");
rxplus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        rxplus_actionPerformed(e);
    }
});
rxminus.setFont(new Font("Dialog", 1, 20));
rxminus.setLabel("-");
rxminus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        rxminus_actionPerformed(e);
    }
});
ryminus.setFont(new Font("Dialog", 1, 20));
ryminus.setLabel("-");
ryminus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ryminus_actionPerformed(e);
    }
});
ryplus.setFont(new Font("Dialog", 1, 20));
ryplus.setLabel("+");
ryplus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ryplus_actionPerformed(e);
    }
});

```

```

));
rzminus.setFont(new Font("Dialog", 1, 20));
rzminus.setLabel("-");
rzminus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        rzminus_actionPerformed(e);
    }
});
rzplus.setFont(new Font("Dialog", 1, 20));
rzplus.setLabel("+");
rzplus.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        rzplus_actionPerformed(e);
    }
});
configuration.setLabel("Configuration");
configuration.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        configuration_actionPerformed(e);
    }
});
ty.setText("textField1");
ty.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ty_actionPerformed(e);
    }
});
tx.setText("textField1");
tx.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        tx_actionPerformed(e);
    }
});
tz.setText("textField1");
tz.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        tz_actionPerformed(e);
    }
});
p_inc.setText("textField1");
p_inc.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        p_inc_actionPerformed(e);
    }
});
rxz.setText("textField1");
rxz.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        rxz_actionPerformed(e);
    }
});
rxy.setText("textField1");
rxy.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)

```



```

    {
        rxy_actionPerformed(e);
    }
});
rxx.setText("textField1");
rxx.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        rxx_actionPerformed(e);
    }
});
rxa.setText("textField1");
rxa.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        rxa_actionPerformed(e);
    }
});
a_inc.setText("textField1");
a_inc.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        a_inc_actionPerformed(e);
    }
});
label1.setFont(new Font("Dialog", 1, 18));
label1.setAlignment(1);
label1.setText("X");
label2.setFont(new Font("Dialog", 1, 18));
label2.setAlignment(1);
label2.setText("Y");
label3.setFont(new Font("Dialog", 1, 18));
label3.setAlignment(1);
label3.setText("Z");
label6.setAlignment(1);
label6.setText("Axis");
label7.setAlignment(1);
label7.setText("Angle (Degree)");
label8.setAlignment(1);
label8.setText("X-axis Rotation");
label9.setAlignment(1);
label9.setText("Y-axis Rotation");
label10.setAlignment(1);
label10.setText("Z-axis Rotation");
label11.setAlignment(1);
label11.setText("Angle Increment");
label4.setForeground(Color.blue);
label4.setAlignment(1);
label4.setFont(new java.awt.Font("Serif", 1, 20));
label4.setText("Translation");
button15.setLabel("button1");
label5.setForeground(Color.blue);
label5.setFont(new java.awt.Font("Serif", 1, 20));
label5.setAlignment(1);
label5.setText("Rotation");
label12.setAlignment(1);
label12.setText("Position Increment");
jacobian.setLabel("Jacobian Matrix");
jacobian.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        jacobian_actionPerformed(e);
    }
});
this.setLayout(xYLayout1);
reserved.setLabel("Get DetJm");
reserved.addActionListener(new java.awt.event.ActionListener()

```

```

{
    public void actionPerformed(ActionEvent e)
    {
        reserved_actionPerformed(e);
    }
});

motion_reset.setLabel("Motion Reset");
checkboxbox1.setLabel("Enable Connector Length Change");
checkboxbox1.addItemListener(new java.awt.event.ItemListener()
{
    public void itemStateChanged(ItemEvent e)
    {
        checkbox1_itemStateChanged(e);
    }
});

this.add(rzplus, new XYConstraints(187, 284, 49, 24));
this.add(rzminus, new XYConstraints(126, 284, 49, 24));
this.add(ryplus, new XYConstraints(187, 256, 49, 24));
this.add(ryminus, new XYConstraints(126, 256, 49, 24));
this.add(rxminus, new XYConstraints(126, 228, 49, 24));
this.add(rxplus, new XYConstraints(187, 228, 49, 24));
this.add(tzplus, new XYConstraints(183, 89, 53, 24));
this.add(tzminus, new XYConstraints(52, 89, 53, 24));
this.add(typlus, new XYConstraints(183, 62, 53, 24));
this.add(tyminus, new XYConstraints(52, 62, 53, 24));
this.add(txplus, new XYConstraints(183, 34, 53, 24));
this.add(txminus, new XYConstraints(52, 34, 53, 24));
this.add(tx, new XYConstraints(111, 34, 65, 24));
this.add(a_inc, new XYConstraints(171, 311, 65, 24));
this.add(rxa, new XYConstraints(145, 200, 91, 24));
this.add(rxx, new XYConstraints(41, 171, 63, 24));
this.add(rxy, new XYConstraints(107, 171, 63, 24));
this.add(rzx, new XYConstraints(173, 171, 63, 24));
this.add(p_inc, new XYConstraints(171, 117, 65, 24));
this.add(tz, new XYConstraints(111, 89, 65, 24));
this.add(ty, new XYConstraints(111, 62, 65, 24));
this.add(label6, new XYConstraints(9, 171, 33, 24));
this.add(label3, new XYConstraints(10, 89, 27, 24));
this.add(label2, new XYConstraints(10, 62, 27, 24));
this.add(label1, new XYConstraints(10, 34, 27, 24));
this.add(label12, new XYConstraints(10, 117, 154, 24));
this.add(label11, new XYConstraints(10, 311, 154, 24));
this.add(label7, new XYConstraints(10, 200, 124, 24));
this.add(label10, new XYConstraints(10, 284, 104, 24));
this.add(label9, new XYConstraints(10, 256, 104, 24));
this.add(label8, new XYConstraints(10, 228, 104, 24));
this.add(label4, new XYConstraints(10, 6, 226, 24));
this.add(button15, new XYConstraints(428, 285, 38, 24));
this.add(label5, new XYConstraints(10, 145, 225, 24));
this.add(motion_reset, new XYConstraints(10, 341, 226, 24));
this.add(configuration, new XYConstraints(10, 388, 226, 24));
this.add(jacobian, new XYConstraints(10, 418, 226, 24));
this.add(reserved, new XYConstraints(10, 448, 226, 24));
this.add(checkboxbox1, new XYConstraints(10, 488, 226, 24));
motion_reset.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        motion_reset_actionPerformed(e);
    }
});
});

public void start()
{
    // getting the VRML browser
    vrmlbrowser = Browser.getBrowser(this);
}

```

```

try {
    // establish the connection between VRML and Java applet
    xfscript = vrmlbrowser.getNode("EAI_CONTROL*");
    xfplatform = vrmlbrowser.getNode("TOP_PLATFORM*");
    for (int i=0; i<6; i++)
    {
        legnodes[i] = vrmlbrowser.getNode("T_LEG"+Integer.toString(i+1));
        legnodes[i+6] = vrmlbrowser.getNode("B_LEG"+Integer.toString(i+1));
    }
    trn = (EventInSFVec3f) xfscript.getEventIn("set_translation");
    rot = (EventInSFRotation) xfscript.getEventIn("set_rotation");
    rst = (EventInSFVec3f) xfscript.getEventIn("set_motion_reset");
    stt = (EventInSFFloat) xfscript.getEventIn("set_triangular_top_set");
    stb = (EventInSFFloat) xfscript.getEventIn("set_triangular_base_set");
    sht = (EventInSFVec3f) xfscript.getEventIn("set_hexagonal_top_set");
    shb = (EventInSFVec3f) xfscript.getEventIn("set_hexagonal_base_set");
    sct = (EventInSFBool) xfscript.getEventIn("set_custom_top_set");
    scb = (EventInSFBool) xfscript.getEventIn("set_custom_base_set");
    iprot = (EventInSFRotation) xfplatform.getEventIn("set_rotation");
    oprot = (EventOutSFRotation) xfplatform.getEventOut("rotation_changed");
    eclc = (EventInSFBool) xfscript.getEventIn("set_connector_length");
    stt.setValue(pdata.getTopSize());
    stb.setValue(pdata.getBaseSize());
    rst.setValue(pdata.getMaxpos());
    eclc.setValue(checkbox1.getState());
}
catch (InvalidNodeException ne) {
    System.out.println("PROBLEMS!: " + ne);
}
catch (InvalidEventInException ee) {
    System.out.println("PROBLEMS!: " + ee);
}
catch (InvalidEventOutException ee) {
    System.out.println("PROBLEMS!: " + ee);
}
}

//Get Applet information

public String getAppletInfo()
{
    return "Applet Information";
}

//Get parameter info

public String[][] getParameterInfo()
{
    return null;
}

// event handler for -x translation button
void txminus_actionPerformed(ActionEvent e)
{
    float[] tval= new float[3];
    float x,y,z;

    Float jFloat;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    z=jFloat.floatValue();
    tval[0]=x-p_incmt;
    tval[1]=y;
    tval[2]=z;
    trn.setValue(tval);
    tx.setText(Float.toString(tval[0]));
}

```

```

// event handler for -y translation button
void tyminus_actionPerformed(ActionEvent e)
{
    float[] tval= new float[3];
    float x,y,z;
    Float jFloat;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    z=jFloat.floatValue();
    tval[0]=x;
    tval[1]=y-p_incmnt;
    tval[2]=z;
    trn.setValue(tval);
    ty.setText(Float.toString(tval[1]));
}

// event handler for +y translation button
void typlus_actionPerformed(ActionEvent e)
{
    float[] tval= new float[3];
    Float jFloat;
    float x,y,z;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    z=jFloat.floatValue();
    tval[0]=x;
    tval[1]=y+p_incmnt;
    tval[2]=z;
    trn.setValue(tval);
    ty.setText(Float.toString(tval[1]));
}

// event handler for +x translation button
void txplus_actionPerformed(ActionEvent e)
{
    float[] tval= new float[3];
    Float jFloat;
    float x,y,z;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    z=jFloat.floatValue();
    tval[0]=x+p_incmnt;
    tval[1]=y;
    tval[2]=z;
    trn.setValue(tval);
    tx.setText(Float.toString(tval[0]));
}

// event handler for +z translation button
void tzplus_actionPerformed(ActionEvent e)
{
    float[] tval= new float[3];
    Float jFloat;
    float x,y,z;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    z=jFloat.floatValue();
    tval[0]=x;

```

```

tval[1]=y;
tval[2]=z+p_incmnt;
trn.setValue(tval);
tz.setText(Float.toString(tval[2]));
}

```

```

// event handler for -z translation button
void tzminus_actionPerformed(ActionEvent e)

```

```

{
    float[] tval= new float[3];
    Float jFloat;
    float x,y,z;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    z=jFloat.floatValue();
    tval[0]=x;
    tval[1]=y;
    tval[2]=z-p_incmnt;
    trn.setValue(tval);
    tz.setText(Float.toString(tval[2]));
}

```

```

// event handler for x translation textbox
void tx_actionPerformed(ActionEvent e)

```

```

{
    float[] tval= new float[3];
    Float jFloat;
    float x,y,z;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    z=jFloat.floatValue();
    tval[0]=x;
    tval[1]=y;
    tval[2]=z;
    trn.setValue(tval);
}

```

```

// event handler for y translation textbox
void ty_actionPerformed(ActionEvent e)

```

```

{
    float[] tval= new float[3];
    Float jFloat;
    float x,y,z;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    z=jFloat.floatValue();
    tval[0]=x;
    tval[1]=y;
    tval[2]=z;
    trn.setValue(tval);
}

```

```

// event handler for z translation textbox
void tz_actionPerformed(ActionEvent e)

```

```

{
    float[] tval= new float[3];
    Float jFloat;
    float x,y,z;
    jFloat=Float.valueOf(tx.getText());
    x=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    y=jFloat.floatValue();
}

```

```

jFloat=Float.valueOf(tz.getText());
z=jFloat.floatValue();
tval[0]=x;
tval[1]=y;
tval[2]=z;
trn.setValue(tval);
}

// event handler for translation increment value textbox
void p_inc_actionPerformed(ActionEvent e)
{
    Float jFloat=Float.valueOf(p_inc.getText());
    p_incmnt=jFloat.floatValue();
    if (p_incmnt<0.0) p_incmnt=1.0f;
    if (p_incmnt>100.0) p_incmnt=1.0f;
    p_inc.setText(Float.toString(p_incmnt));
}

// event handler for rotation axis textbox (x)
void rxn_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    Float jFloat;
    jFloat=Float.valueOf(rxx.getText());
    rval[0]=jFloat.floatValue();
    jFloat=Float.valueOf(rxy.getText());
    rval[1]=jFloat.floatValue();
    jFloat=Float.valueOf(rxz.getText());
    rval[2]=jFloat.floatValue();
    jFloat=Float.valueOf(rxa.getText());
    rval[3]=jFloat.floatValue()*0.017453f;
    iprot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

// event handler for rotation axis textbox (y)
void rxy_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    Float jFloat;
    jFloat=Float.valueOf(rxx.getText());
    rval[0]=jFloat.floatValue();
    jFloat=Float.valueOf(rxy.getText());
    rval[1]=jFloat.floatValue();
    jFloat=Float.valueOf(rxz.getText());
    rval[2]=jFloat.floatValue();
    jFloat=Float.valueOf(rxa.getText());
    rval[3]=jFloat.floatValue()*0.017453f;
    iprot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

// event handler for rotation axis textbox (z)
void rxz_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    Float jFloat;
    jFloat=Float.valueOf(rxx.getText());
    rval[0]=jFloat.floatValue();
    jFloat=Float.valueOf(rxy.getText());
    rval[1]=jFloat.floatValue();
    jFloat=Float.valueOf(rxz.getText());
    rval[2]=jFloat.floatValue();
    jFloat=Float.valueOf(rxa.getText());
    rval[3]=jFloat.floatValue()*0.017453f;
    iprot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

```

```

// event handler for rotation angle textbox
void rxa_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    Float jFloat;
    jFloat=Float.valueOf(rxx.getText());
    rval[0]=jFloat.floatValue();
    jFloat=Float.valueOf(rxy.getText());
    rval[1]=jFloat.floatValue();
    jFloat=Float.valueOf(rxz.getText());
    rval[2]=jFloat.floatValue();
    jFloat=Float.valueOf(rxa.getText());
    rval[3]=jFloat.floatValue()*0.017453f;
    iprot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

// event handler for -x rotation button
void rxminus_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    rval[0]=1;
    rval[1]=0;
    rval[2]=0;
    rval[3]=-a_incmt*0.017453f;
    rot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

// event handler for +x rotation button
void rxplus_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    rval[0]=1;
    rval[1]=0;
    rval[2]=0;
    rval[3]=a_incmt*0.017453f;
    rot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

// event handler for +y rotation button
void ryplus_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    rval[0]=0;
    rval[1]=1;
    rval[2]=0;
    rval[3]=a_incmt*0.017453f;
    rot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

// event handler for +z rotation button
void rzplus_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    rval[0]=0;
    rval[1]=0;
    rval[2]=1;
    rval[3]=a_incmt*0.017453f;
    rot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

```

```

// setting data of the rotation axis and angle textboxes
void set_rotation_text(float[] rval)
{
    rx.setText(Float.toString((float)Math.ceil((double)rval[0]*1000)/1000));
    ry.setText(Float.toString((float)Math.ceil((double)rval[1]*1000)/1000));
    rz.setText(Float.toString((float)Math.ceil((double)rval[2]*1000)/1000));
    rx.setText(Float.toString((float)Math.ceil((double)rval[3]*57.29578*100/100));
}

// event handler for rotation angle increment textbox
void a_inc_actionPerformed(ActionEvent e)
{
    Float jFloat=Float.valueOf(a_inc.getText());
    a_incm=jFloat.floatValue();
    if (a_incm<0.0) a_incm=1.0f;
    if (a_incm>90.0) a_incm=1.0f;
    a_inc.setText(Float.toString(a_incm));
}

// event handler for motion reset button
void motion_reset_actionPerformed(ActionEvent e)
{
    float[] rval = new float[4];
    rst.setValue(pdata.getMaxpos());
    tx.setText(Float.toString(pdata.getMaxpos()[0]));
    ty.setText(Float.toString(pdata.getMaxpos()[1]));
    tz.setText(Float.toString(pdata.getMaxpos()[2]));
    rval[0]=0.0f; rval[1]=0.0f; rval[2]=1.0f; rval[3]=0.0f;
    iprot.setValue(rval);
    rval[0]=1.0f; rval[1]=0.0f; rval[2]=0.0f; rval[3]=pdata.getMaxrot()[0]*0.017453f;
    rot.setValue(rval);
    rval[0]=0.0f; rval[1]=1.0f; rval[2]=0.0f; rval[3]=pdata.getMaxrot()[1]*0.017453f;
    rot.setValue(rval);
    rval[0]=0.0f; rval[1]=0.0f; rval[2]=1.0f; rval[3]=pdata.getMaxrot()[2]*0.017453f;
    rot.setValue(rval);
    rval=oprot.getValue();
    set_rotation_text(rval);
}

// event handler for configuration button
void configuration_actionPerformed(ActionEvent e)
{
    if (config==null || !config.isValid())
    {
        config = new Config_Dialog(new Frame(),"Configuration",
                                   true, pdata);

        config.stt = stt;
        config.stb = stb;
        config.sht = sht;
        config.shb = shb;
        config.sct = sct;
        config.scb = scb;

        config.show();
    }
}

// event handler for -y rotation button
void ryminus_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    rval[0]=0;
    rval[1]=1;
    rval[2]=0;
    rval[3]=-a_incm*0.017453f;
    rot.setValue(rval);
    rval=oprot.getValue();
    rx.setText(Float.toString((float)Math.round(rval[0]*1000)/1000f));
    ry.setText(Float.toString((float)Math.round(rval[1]*1000)/1000f));
    rz.setText(Float.toString((float)Math.round(rval[2]*1000)/1000f));
    rx.setText(Float.toString((float)Math.round(rval[3]*57.29578f*1000)/1000f));
}

```



```

}

// event handler for -z rotation button
void rzminus_actionPerformed(ActionEvent e)
{
    float[] rval= new float[4];
    rval[0]=0;
    rval[1]=0;
    rval[2]=1;
    rval[3]=-a_incmnt*0.017453f;
    rot.setValue(rval);
    rval=oprot.getValue();
    rxx.setText(Float.toString((float)Math.round(rval[0]*1000)/1000f));
    rxy.setText(Float.toString((float)Math.round(rval[1]*1000)/1000f));
    rxz.setText(Float.toString((float)Math.round(rval[2]*1000)/1000f));
    rxa.setText(Float.toString((float)Math.round(rval[3]*57.29578f*1000)/1000f));
}

// event handler for Jacobian matrix button
void jacobian_actionPerformed(ActionEvent e)
{
    float[][] jmat;// 6x6 matrix
    float detj, qi;

    jmat = get_jacobian();
    detj = get_determinant(jmat);
    if (pdata.getMaxdetFound()) qi = Math.abs(detj)/pdata.getMaxdet();
    else qi = -1.0f;
    //System.out.println(Djacobian);
    if (Djacobian==null || !Djacobian.isValid())
    {
        Djacobian = new JB_Dialog(new Frame(),"Jacobian Matrix",
                                   true, jmat, detj, qi);
        Djacobian.show();
    }
}

// event handler for Get DetJm button
void reserved_actionPerformed(ActionEvent e)
{
    // connect to the server
    float[][] p = new float[12][3];
    float[][] q;//= new float[12][3];
    float l;
    float[] maxpos = new float[3];
    float[] maxrot = new float[3];
    MessageDialog message;
    EventOutSFVec3f gnd = null;
    String result, retval;
    String messagetext;

    if (pdata.getMaxdetFound())
    {
        messagetext="The DetJm of the platform is already calculated.";
        message = new MessageDialog(new Frame(),"Message Dialog", true, messagetext);
        message.show();
        return;
    }

    // generating unique(untitized) configuration data for the platform

    for (int i=0; i<12; i++)
    {
        gnd = (EventOutSFVec3f)legnodes[i].getEventOut("translation_changed");
        p[i] = gnd.getValue();
    }

    l=(float)Math.sqrt(p[0][0]*p[0][0]+p[0][1]*p[0][1]);

    System.out.println("scalar = " + l);
}

```

```

String topconfig = "";
String baseconfig = "";

for(int i=0; i<6; i++)
{
    topconfig+= Long.toString((long)Math.ceil(((double) (p[i][0]/1)*10000.0))+"x"
        +Long.toString((long)Math.ceil(((double) (p[i][1]/1)*10000.0))+"x");
    baseconfig+=Long.toString((long)Math.ceil(((double) (p[i+6][0]/1)*10000.0))+"x"
        +Long.toString((long)Math.ceil(((double) (p[i+6][1]/1)*10000.0))+"x");
}

// server URL and the ASP page location
String qiurl = "http://localhost/QINB/QIDB.ASP?top=" + topconfig
    + "&base=" + baseconfig;

// connect to the server
try {
    URL maxdet = new URL(qiurl);
    URLConnection uc = maxdet.openConnection();
    BufferedReader in = new BufferedReader(
        new InputStreamReader(
            uc.getInputStream()));

    // return value from the server
    result = in.readLine();
    if (result.equals("FOUND"))
    {
        pdata.setMaxdet(Float.valueOf(in.readLine()).floatValue()*1*1*1);

        retval = in.readLine();
        maxpos[0] = Float.valueOf(retval).floatValue()*1;
        retval = in.readLine();
        maxpos[1] = Float.valueOf(retval).floatValue()*1;
        retval = in.readLine();
        maxpos[2] = Float.valueOf(retval).floatValue()*1;

        retval = in.readLine();
        maxrot[0] = Float.valueOf(retval).floatValue();
        retval = in.readLine();
        maxrot[1] = Float.valueOf(retval).floatValue();
        retval = in.readLine();
        maxrot[2] = Float.valueOf(retval).floatValue();

        pdata.setMaxpos(maxpos);
        pdata.setMaxrot(maxrot);
        pdata.setMaxdetFound(true);

        messagetext="The DetJm of the platform found.\n"
            +"DetJm = "+Float.toString(pdata.getMaxdet())+"\n"
            +"Pxzy = "+Float.toString(maxpos[0])+" "
            +Float.toString(maxpos[1])+" "
            +Float.toString(maxpos[2])+"\n"
            +"Rxyz = "+Float.toString(maxrot[0])+" "
            +Float.toString(maxrot[1])+" "
            +Float.toString(maxrot[2]);

        // found dialog
        message = new MessageDialog(new Frame(),"Message Dialog", true, messagetext);
        message.show();
    }

    if (result.equals("PROCESSING"))
    {
        // processing dialog
        pdata.setMaxdetFound(false);
        messagetext="The DetJm of the platform is not on the database.\n"
            +"The search program has started on the server.\n"
            +"Please try later to get the result.";
        message = new MessageDialog(new Frame(),"Message Dialog", true, messagetext);
        message.show();
    }
}

```

```

if (result.equals("WAIT"))
{
    // wait dialog
    pdata.setMaxdetFound(false);
    messagetext="The DetJm of the platform is not on the database.\n"
        +"Another search program is already running on the server.\n"
        +"Only one process of the search program is allowed.\n"
        +"Please try again later to start the search program.";
    message = new MessageDialog(new Frame(), "Message Dialog", true, messagetext);
    message.show();
}

in.close();
}
catch (MalformedURLException ne) {
    System.out.println("Error : " + ne);
}
catch (IOException ne) {
    System.out.println("Error : " + ne);
}
}

// get Jacobian matrix
float[][] get_jacobian()
{
    float[][] jmat = new float[6][6];
    float[][] p ;//= new float[12][3];
    float l;

    p = get_joint_coords();

    for (int i=0; i<6; i++)
    {
        l = (float)Math.sqrt((p[i][0]-p[i+6][0])*(p[i][0]-p[i+6][0])
            +(p[i][1]-p[i+6][1])*(p[i][1]-p[i+6][1])+(p[i][2]-p[i+6][2])
            *(p[i][2]-p[i+6][2]));
        jmat[0][i] = (p[i][0]-p[i+6][0])/l;
        jmat[1][i] = (p[i][1]-p[i+6][1])/l;
        jmat[2][i] = (p[i][2]-p[i+6][2])/l;
        jmat[3][i] = (p[i][1]*p[i+6][2]-p[i][2]*p[i+6][1])/l;
        jmat[4][i] = (p[i][2]*p[i+6][0]-p[i][0]*p[i+6][2])/l;
        jmat[5][i] = (p[i][0]*p[i+6][1]-p[i][1]*p[i+6][0])/l;
    }

    return jmat;
}

// get determinant of J
float get_determinant(float[][] J)
{
    float ans=0;
    float[] s = new float[6];
    float[][] D = new float[5][6];
    int i,j,k,l,m,n;

    s[0]=0;
    for(i=0;i<6;i++)
    {
        s[1]=0;
        D[0][i]=0;
        for(j=0;j<6;j++)
        {
            if(j==i) continue;
            s[2]=0;
            D[1][j]=0;
            for(k=0;k<6;k++)
            {
                if(k==i||k==j) continue;
                s[3]=0;

```

```

D[2][k]=0;
for(l=0;l<6;l++)
{
    if(l==i||l==j||l==k) continue;
    s[4]=0;
    D[3][l]=0;
    for(m=0;m<6;m++)
    {
        if(m==i||m==j||m==k||m==l) continue;
        s[5]=0;
        D[4][m]=0;
        for(n=0;n<6;n++)
        {
            if(n==i||n==j||n==k||n==l||n==m) continue;
            D[4][m]=(float)Math.pow(-1,s[5])*J[5][n];
            s[5]++;
        }
        D[3][l]+=(float)Math.pow(-1,s[4])*J[4][m]*D[4][m];
        s[4]++;
    }
    D[2][k]+=(float)Math.pow(-1,s[3])*J[3][l]*D[3][l];
    s[3]++;
}
D[1][j]+=(float)Math.pow(-1,s[2])*J[2][k]*D[2][k];
s[2]++;
}
D[0][i]+=(float)Math.pow(-1,s[1])*J[1][j]*D[1][j];
s[1]++;
}
ans+=(float)Math.pow(-1,s[0])*J[0][i]*D[0][i];
s[0]++;
}
return Math.abs(ans);
}

// get joint coordinates
float[] get_joint_coords()
{
    float[][] p = new float[12][3];
    float[] tval= new float[3];
    float[][] mat = new float[4][4];
    double[] m_axis= new double[4];
    double angle;
    float[] ppp = new float[3];
    EventOutSFVec3f gnd = null;
    Float jFloat;
    Double jDouble;

    jDouble=Double.valueOf(rxx.getText());
    m_axis[0]=jDouble.doubleValue();
    jDouble=Double.valueOf(rxy.getText());
    m_axis[1]=jDouble.doubleValue();
    jDouble=Double.valueOf(rxz.getText());
    m_axis[2]=jDouble.doubleValue();
    jDouble=Double.valueOf(rxa.getText());
    angle=jDouble.doubleValue()*0.017453;

    jFloat=Float.valueOf(tx.getText());
    tval[0]=jFloat.floatValue();
    jFloat=Float.valueOf(ty.getText());
    tval[1]=jFloat.floatValue();
    jFloat=Float.valueOf(tz.getText());
    tval[2]=jFloat.floatValue();

    // generating current position and orientation of the platform in matrix form
    mat[0][0] = (float) (m_axis[0]*m_axis[0]*(1.0-Math.cos(angle))+Math.cos(angle));
    mat[1][0] = (float) (m_axis[0]*m_axis[1]*(1.0-Math.cos(angle))
        +m_axis[2]*Math.sin(angle));
    mat[2][0] = (float) (m_axis[0]*m_axis[2]*(1.0-Math.cos(angle))
        -m_axis[1]*Math.sin(angle));

```

```

mat[3][0] = 0.0f;
mat[0][1] = (float)(m_axis[1]*m_axis[0]*(1.0-Math.cos(angle))
               -m_axis[2]*Math.sin(angle));
mat[1][1] = (float)(m_axis[1]*m_axis[1]*(1.0-Math.cos(angle))
               +Math.cos(angle));
mat[2][1] = (float)(m_axis[1]*m_axis[2]*(1.0-Math.cos(angle))
               +m_axis[0]*Math.sin(angle));
mat[3][1] = 0.0f;
mat[0][2] = (float)(m_axis[2]*m_axis[0]*(1.0-Math.cos(angle))
               +m_axis[1]*Math.sin(angle));
mat[1][2] = (float)(m_axis[2]*m_axis[1]*(1.0-Math.cos(angle))
               -m_axis[0]*Math.sin(angle));
mat[2][2] = (float)(m_axis[2]*m_axis[2]*(1.0-Math.cos(angle))
               +Math.cos(angle));
mat[3][2] = 0.0f;
mat[0][3] = tval[0];
mat[1][3] = tval[1];
mat[2][3] = tval[2];
mat[3][3] = 1.0f;

// getting the coordinates of the joints of the platform (initial position)
for (int i=0; i<12; i++)
{
    gnd = (EventOutSFVec3f)legnodes[i].getEventOut("translation_changed");
    p[i] = gnd.getValue();
}

// multiply current transformation matrix to the initial joint coordinates
for (int i=0; i<6; i++) p[i]=matrix_multiply(mat,p[i]);

return p;
}

// matrix-vector multiplication (4x4, 4x1)
float[] matrix_multiply(float[][] mat, float p[])
{
    float[] pp = new float[4];
    float[] ans = new float[4];
    float[] ret = new float[3];

    for (int i=0; i<3; i++) pp[i]=p[i];
    pp[3]=1.0f;

    for(int i=0; i<4; i++)
    {
        ans[i] = 0.0f;
        for(int j=0; j<4; j++)
            ans[i]+=mat[i][j]*pp[j];
    }

    for (int i=0; i<3; i++)
        ret[i]=ans[i];

    return ret;
}

// event handler for the "enable leg length change" checkbox
void checkbox1_itemStateChanged(ItemEvent e)
{
    eelc.setValue(checkbox1.getState());
}
}

```

## 5. MessageDialog.java

```
//Title:      VRML Platform
//Copyright:  Copyright (c) 2000
//Author:     Jaehoon Lee

package Platform;

import java.awt.*;
import java.awt.event.*;
import com.borland.jbcl.layout.*;

/*****
 * Message display dialog class
 *****/

public class MessageDialog extends Dialog
{
    Panel panell = new Panel();
    XYLayout xYLayout1 = new XYLayout();
    Button OK = new Button();
    TextArea textAreal = new TextArea("Message from Server",4,1,TextArea.SCROLLBARS_NONE);

    // constructor
    public MessageDialog(Frame frame, String title, boolean modal, String text)
    {
        super(frame, title, modal);
        enableEvents(AWTEvent.WINDOW_EVENT_MASK);
        try
        {
            jbInit();
            add(panell);
            pack();
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
        textAreal.setText(text);
    }

    public MessageDialog(Frame frame)
    {
        this(frame, "", false, "");
    }

    public MessageDialog(Frame frame, boolean modal)
    {
        this(frame, "", modal, "");
    }

    public MessageDialog(Frame frame, String title)
    {
        this(frame, title, false, "");
    }

    void jbInit() throws Exception
    {
        panell.setLayout(xYLayout1);
        OK.setName("OK");
        OK.setLabel("OK");
        OK.addActionListener(new java.awt.event.ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                OK_actionPerformed(e);
            }
        });
    }
}
```

```

xYLayout1.setHeight(156);
xYLayout1.setWidth(374);
textAreal.setEditable(false);
panell.add(textAreal, new XYConstraints(11, 12, 351, 90));
panell.add(OK, new XYConstraints(125, 115, 125, 26));
}

protected void processWindowEvent(WindowEvent e)
{
    if(e.getID() == WindowEvent.WINDOW_CLOSING)
    {
        cancel();
    }
    super.processWindowEvent(e);
}

void cancel()
{
    dispose();
}

void OK_actionPerformed(ActionEvent e)
{
    dispose();
}
}

```

## 6. PLATFORM.WRL

```

#VRML V2.0 utf8

#Title:          VRML Platform
#Copyright:      Copyright (c) 2000
#Author:         Jaehoon Lee

NavigationInfo {type [ "EXAMINE" ]}

Background { skyColor 0 0 1 }

DEF VIEW1 Viewpoint {
    orientation 1 0 0 1
    position 0 -50 35
    description "Home"
}

DEF VIEW2 Viewpoint {
    orientation 1 0 0 1.5708
    position 0 -50 9
    description "Side"
}

DEF VIEW3 Viewpoint {
    orientation 0 0 1 0
    position 0 0 50
    description "Top"
}

DEF BASE_PLATFORM Transform {
    children [
        Transform {
            children [
                Shape {
                    appearance Appearance {
                        material DEF BASE_PLATE_COLOR Material {
                            diffuseColor 0 0.5 0.5
                            transparency 0.3
                        }
                    }
                }
            ]
        }
    ]
}

```

```

geometry IndexedFaceSet(
  coord DEF BASE_PLATE Coordinate (
    point [ 17.3205 -10 -.3,
            0 20 -.3,
            0 20 -.3,
            -17.3205 -10 -.3,
            -17.3205 -10 -.3,
            17.3205 -10 -.3,
            17.3205 -10 .3,
            0 20 .3,
            0 20 .3,
            -17.3205 -10 .3,
            -17.3205 -10 .3,
            17.3205 -10 .3,
            0 0 0
          ]
    )
  coordIndex [0,12,1,-1,
              1,12,2,-1,
              2,12,3,-1,
              3,12,4,-1,
              4,12,5,-1,
              5,12,0,-1,
              11,12,10,-1,
              10,12,9,-1,
              9,12,8,-1,
              8,12,7,-1,
              7,12,6,-1,
              6,12,11,-1,
              0,1,7,6,-1,
              1,2,8,7,-1,
              2,3,9,8,-1,
              3,4,10,9,-1,
              4,5,11,10,-1,
              5,0,6,11,-1
            ]
  )
)
)
)
DEF B_LEG1 Transform (
  translation 17.3205 -10 0
  rotation 0 0 1 0
  children (
    Transform (
      children (
        Shape (
          appearance Appearance (
            material Material (
              diffuseColor 1.0 0.0 0.0
            )
          )
          geometry Sphere (
            radius 1
          )
        )
      )
      DEF TS_B1 PlaneSensor (
        offset 17.3205 -10 0
        enabled FALSE
      )
    )
  )
)
DEF B_LEG1A Transform (
  rotation 0 0 1 2.0944
  children (
    DEF B_LEG1B Transform (
      rotation 0 -1 0 0.7854
      children (
        DEF BASE_LEG Transform (
          translation 7.5 0 0 # half of the height
          rotation 0 0 1 -1.5708
        )
      )
    )
  )
)

```



[illegible]

```

        children [
            DEF B_LEG2B Transform {
                rotation 0 -1 0 0.7854
                children [
                    USE BASE_LEG
                    DEF T_LEG2A Transform {
                        translation 14.4949 0 0
                        rotation 0 0 1 -1.5708
                        scale 0.25 20 0.25
                        children [
                            USE TOP_LEG_BAR
                            USE TLEG_SENSOR
                        ]
                    }
                ]
            }
        ]
    ]
}

DEF B_LEG3 Transform {
    translation 0 20 0
    rotation 0 0 1 0
    children [
        Transform {
            children [
                USE BASE_LEG_BALL
                DEF TS_B3 PlaneSensor {
                    offset 0 20 0
                    enabled FALSE
                }
            ]
        }
        DEF B_LEG3A Transform {
            rotation 0 0 1 -2.0944
            children [
                DEF B_LEG3B Transform {
                    rotation 0 -1 0 0.7854
                    children [
                        USE BASE_LEG
                        DEF T_LEG3A Transform {
                            translation 14.4949 0 0
                            rotation 0 0 1 -1.5708
                            scale 0.25 20 0.25
                            children [
                                USE TOP_LEG_BAR
                                USE TLEG_SENSOR
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}

DEF B_LEG4 Transform {
    translation -17.3205 -10 0
    rotation 0 0 1 0
    children [
        Transform {
            children [
                USE BASE_LEG_BALL
                DEF TS_B4 PlaneSensor {
                    offset -17.3205 -10 0
                    enabled FALSE
                }
            ]
        }
    ]
    DEF B_LEG4A Transform {

```

```

rotation 0 0 1 1.0472
children [
  DEF B_LEG4B Transform {
    rotation 0 -1 0 0.7854
    children [
      USE BASE_LEG
      DEF T_LEG4A Transform {
        translation 14.4949 0 0
        rotation 0 0 1 -1.5708
        scale 0.25 20 0.25
        children [
          USE TOP_LEG_BAR
          USE TLEG_SENSOR
        ]
      }
    ]
  }
]
}

DEF B_LEG5 Transform {
  translation -17.3205 -10 0
  rotation 0 0 1 0
  children [
    Transform {
      children [
        USE BASE_LEG_BALL
        DEF TS_B5 PlaneSensor {
          offset -17.3205 -10 0
          enabled FALSE
        }
      ]
    }
  ]
}

DEF B_LEG5A Transform {
  rotation 0 0 1 0
  children [
    DEF B_LEG5B Transform {
      rotation 0 -1 0 0.7854
      children [
        USE BASE_LEG
        DEF T_LEG5A Transform {
          translation 14.4949 0 0
          rotation 0 0 1 -1.5708
          scale 0.25 20 0.25
          children [
            USE TOP_LEG_BAR
            USE TLEG_SENSOR
          ]
        }
      ]
    }
  ]
}

DEF B_LEG6 Transform {
  translation 17.3205 -10 0
  rotation 0 0 1 0
  children [
    Transform {
      children [
        USE BASE_LEG_BALL
        DEF TS_B6 PlaneSensor {
          offset 17.3205 -10 0
          enabled FALSE
        }
      ]
    }
  ]
}

```

```

DEF B_LEG6A Transform {
  rotation 0 0 1 3.1416
  children {
    DEF B_LEG6B Transform {
      rotation 0 1 0 0.7854
      children {
        USE BASE_LEG
        DEF T_LEG6A Transform {
          translation 14.4949 0 0
          rotation 0 0 1 -1.5708
          scale 0.25 20 0.25
          children {
            USE TOP_LEG_BAR
            USE TLEG_SENSOR
          }
        }
      }
    }
  }
}

DEF TOP_PLATFORM Transform {
  translation 0 0 17.3205
  rotation 0 0 1 0
  children {
    Transform {
      children {
        Shape {
          appearance Appearance {
            material Material {
              diffuseColor 0.5 0.5 0.5
              transparency 0.3
            }
          }
        }
        geometry IndexedFaceSet{
          coord DEF TOP_PLATE Coordinate {
            point [ 8.6603 5 -.3,
                    8.6603 5 -.3,
                    -8.6603 5 -.3,
                    -8.6603 5 -.3,
                    0 -10 -.3,
                    0 -10 -.3,
                    8.6603 5 .3,
                    8.6603 5 .3,
                    -8.6603 5 .3,
                    -8.6603 5 .3,
                    0 -10 .3,
                    0 -10 .3,
                    0 0 0
                  ]
          coordIndex [0,12,1,-1,
                    1,12,2,-1,
                    2,12,3,-1,
                    3,12,4,-1,
                    4,12,5,-1,
                    5,12,0,-1,
                    11,12,10,-1,
                    10,12,9,-1,
                    9,12,8,-1,
                    8,12,7,-1,
                    7,12,6,-1,
                    6,12,11,-1,
                    0,1,7,6,-1,
                    1,2,8,7,-1,
                    2,3,9,8,-1,

```

```

3,4,10,9,-1,
4,5,11,10,-1,
5,0,6,11,-1
    ]
  }
}
]
}
DEF T_LEG1 Transform {
  translation 8.6603 5 0
  rotation 0 0 1 0
  children [
    DEF T_LEG1B Transform {
      children [
        Shape {
          appearance Appearance {
            material Material {
              diffuseColor 1.0 0.0 0.0
            }
          }
          geometry Sphere {
            radius 0.5
          }
        }
        DEF TS_T1 PlaneSensor {
          offset 8.6603 5 0
          enabled FALSE
        }
      ]
    }
  ]
}
DEF T_LEG2 Transform {
  translation 8.6603 5 0
  rotation 0 0 1 0
  children [
    DEF T_LEG2B Transform {
      children [
        DEF TOP_LEG_BALL Shape {
          appearance Appearance {
            material DEF TOP_BALL_COLOR Material {
              diffuseColor 0.7 0.7 0.7
            }
          }
          geometry Sphere {
            radius 0.5
          }
        }
        DEF TS_T2 PlaneSensor {
          offset 8.6603 5 0
          enabled FALSE
        }
      ]
    }
  ]
}
DEF T_LEG3 Transform {
  translation -8.6603 5 0
  rotation 0 0 1 0
  children [
    DEF T_LEG3B Transform {
      children [
        USE TOP_LEG_BALL
        DEF TS_T3 PlaneSensor {
          offset -8.6603 5 0
          enabled FALSE
        }
      ]
    }
  ]
}

```

```

DEF T_LEG4 Transform {
  translation -8.6603 5 0
  rotation 0 0 1 0
  children [
    DEF T_LEG4B Transform {
      children [
        USE TOP_LEG_BALL
        DEF TS_T4 PlaneSensor {
          offset -8.6603 5 0
          enabled FALSE
        }
      ]
    }
  ]
}

DEF T_LEG5 Transform {
  translation 0 -10 0
  rotation 0 0 1 0
  children [
    DEF T_LEG5B Transform {
      children [
        USE TOP_LEG_BALL
        DEF TS_T5 PlaneSensor {
          offset 0 -10 0
          enabled FALSE
        }
      ]
    }
  ]
}

DEF T_LEG6 Transform {
  translation 0 -10 0
  rotation 0 0 1 0
  children [
    DEF T_LEG6B Transform {
      children [
        USE TOP_LEG_BALL
        DEF TS_T6 PlaneSensor {
          offset 0 -10 0
          enabled FALSE
        }
      ]
    }
  ]
}

# script node for leg position and orientation changes
DEF LEGS Script{
  directOutput TRUE
  eventIn SFVec3f xyz
  eventIn SFRotation rxyz
  eventIn SFVec3f txyz
  eventIn SFVec3f hxyz
  field SFNode top_platform USE TOP_PLATFORM
  field SFNode top_plate USE TOP_PLATE
  field SFNode base_plate USE BASE_PLATE
  field SFNode bl1 USE B_LEG1
  field SFNode bl1a USE B_LEG1A
  field SFNode bl1b USE B_LEG1B
  field SFNode bl2 USE B_LEG2
  field SFNode bl2a USE B_LEG2A
  field SFNode bl2b USE B_LEG2B
  field SFNode bl3 USE B_LEG3
  field SFNode bl3a USE B_LEG3A
  field SFNode bl3b USE B_LEG3B
  field SFNode bl4 USE B_LEG4

```

```

field SFNode bl4a USE B_LEG4A
field SFNode bl4b USE B_LEG4B
field SFNode bl5 USE B_LEG5
field SFNode bl5a USE B_LEG5A
field SFNode bl5b USE B_LEG5B
field SFNode bl6 USE B_LEG6
field SFNode bl6a USE B_LEG6A
field SFNode bl6b USE B_LEG6B
field SFNode t11 USE T_LEG1
field SFNode t11a USE T_LEG1A
field SFNode t12 USE T_LEG2
field SFNode t12a USE T_LEG2A
field SFNode t13 USE T_LEG3
field SFNode t13a USE T_LEG3A
field SFNode t14 USE T_LEG4
field SFNode t14a USE T_LEG4A
field SFNode t15 USE T_LEG5
field SFNode t15a USE T_LEG5A
field SFNode t16 USE T_LEG6
field SFNode t16a USE T_LEG6A
url *javascript:
function resize_top_plate()
{
    top_plate.point[0]=top_plate.point[6]=t11.translation;
    top_plate.point[0][2]=-0.3;
    top_plate.point[6][2]=0.3;
    top_plate.point[1]=top_plate.point[7]=t12.translation;
    top_plate.point[1][2]=-0.3;
    top_plate.point[7][2]=0.3;
    top_plate.point[2]=top_plate.point[8]=t13.translation;
    top_plate.point[2][2]=-0.3;
    top_plate.point[8][2]=0.3;
    top_plate.point[3]=top_plate.point[9]=t14.translation;
    top_plate.point[3][2]=-0.3;
    top_plate.point[9][2]=0.3;
    top_plate.point[4]=top_plate.point[10]=t15.translation;
    top_plate.point[4][2]=-0.3;
    top_plate.point[10][2]=0.3;
    top_plate.point[5]=top_plate.point[11]=t16.translation;
    top_plate.point[5][2]=-0.3;
    top_plate.point[11][2]=0.3;
    top_plate.point[12][0]=top_plate.point[12][1]=top_plate.point[12][2]=0.0;
}
function resize_base_plate()
{
    base_plate.point[0]=base_plate.point[6]=bl1.translation;
    base_plate.point[0][2]=-0.3;
    base_plate.point[6][2]=0.3;
    base_plate.point[1]=base_plate.point[7]=bl2.translation;
    base_plate.point[1][2]=-0.3;
    base_plate.point[7][2]=0.3;
    base_plate.point[2]=base_plate.point[8]=bl3.translation;
    base_plate.point[2][2]=-0.3;
    base_plate.point[8][2]=0.3;
    base_plate.point[3]=base_plate.point[9]=bl4.translation;
    base_plate.point[3][2]=-0.3;
    base_plate.point[9][2]=0.3;
    base_plate.point[4]=base_plate.point[10]=bl5.translation;
    base_plate.point[4][2]=-0.3;
    base_plate.point[10][2]=0.3;
    base_plate.point[5]=base_plate.point[11]=bl6.translation;
    base_plate.point[5][2]=-0.3;
    base_plate.point[11][2]=0.3;
    base_plate.point[12][0]=base_plate.point[12][1]=base_plate.point[12][2]=0.0;
}
function adjusting()
{
    vecf=top_platform.rotation.multVec(t11.translation);
    dx=vecf[0]+top_platform.translation[0]-bl1.translation[0];
    dy=vecf[1]+top_platform.translation[1]-bl1.translation[1];
    dz=vecf[2]+top_platform.translation[2]-bl1.translation[2];

```

```

b11a.rotation[3]=Math.atan2(dy,dx);
b11b.rotation[3]=Math.atan2(dz,Math.sqrt(dx*dx+dy*dy));
l1len=Math.sqrt(dx*dx+dy*dy+dz*dz);
t11a.translation[0]=l1len-t11a.scale[1]/2;

vecf=top_platform.rotation.multVec(t12.translation);
dx=vecf[0]+top_platform.translation[0]-b12.translation[0];
dy=vecf[1]+top_platform.translation[1]-b12.translation[1];
dz=vecf[2]+top_platform.translation[2]-b12.translation[2];
b12a.rotation[3]=Math.atan2(dy,dx);
b12b.rotation[3]=Math.atan2(dz,Math.sqrt(dx*dx+dy*dy));
l1len=Math.sqrt(dx*dx+dy*dy+dz*dz);
t12a.translation[0]=l1len-t12a.scale[1]/2;

vecf=top_platform.rotation.multVec(t13.translation);
dx=vecf[0]+top_platform.translation[0]-b13.translation[0];
dy=vecf[1]+top_platform.translation[1]-b13.translation[1];
dz=vecf[2]+top_platform.translation[2]-b13.translation[2];
b13a.rotation[3]=Math.atan2(dy,dx);
b13b.rotation[3]=Math.atan2(dz,Math.sqrt(dx*dx+dy*dy));
l1len=Math.sqrt(dx*dx+dy*dy+dz*dz);
t13a.translation[0]=l1len-t13a.scale[1]/2;

vecf=top_platform.rotation.multVec(t14.translation);
dx=vecf[0]+top_platform.translation[0]-b14.translation[0];
dy=vecf[1]+top_platform.translation[1]-b14.translation[1];
dz=vecf[2]+top_platform.translation[2]-b14.translation[2];
b14a.rotation[3]=Math.atan2(dy,dx);
b14b.rotation[3]=Math.atan2(dz,Math.sqrt(dx*dx+dy*dy));
l1len=Math.sqrt(dx*dx+dy*dy+dz*dz);
t14a.translation[0]=l1len-t14a.scale[1]/2;

vecf=top_platform.rotation.multVec(t15.translation);
dx=vecf[0]+top_platform.translation[0]-b15.translation[0];
dy=vecf[1]+top_platform.translation[1]-b15.translation[1];
dz=vecf[2]+top_platform.translation[2]-b15.translation[2];
b15a.rotation[3]=Math.atan2(dy,dx);
b15b.rotation[3]=Math.atan2(dz,Math.sqrt(dx*dx+dy*dy));
l1len=Math.sqrt(dx*dx+dy*dy+dz*dz);
t15a.translation[0]=l1len-t15a.scale[1]/2;

vecf=top_platform.rotation.multVec(t16.translation);
dx=vecf[0]+top_platform.translation[0]-b16.translation[0];
dy=vecf[1]+top_platform.translation[1]-b16.translation[1];
dz=vecf[2]+top_platform.translation[2]-b16.translation[2];
b16a.rotation[3]=Math.atan2(dy,dx);
b16b.rotation[3]=Math.atan2(dz,Math.sqrt(dx*dx+dy*dy));
l1len=Math.sqrt(dx*dx+dy*dy+dz*dz);
t16a.translation[0]=l1len-t16a.scale[1]/2;
}
function xyz(value) { adjusting(); }
function rxyz(value) { adjusting(); }
function txyz(value)
{
    adjusting();
    resize_top_plate();
}

function bxyz(value)
{
    adjusting();
    resize_base_plate();
}
}"

# script for base leg length change (visual)
DEF LEG_LEN_CHANGE Script {
    directOutput TRUE
    eventIn SFVec3f trin
    field SFNode blegtr USE BASE_LEG
    url *javascript:

```



```

function trin(value)
{
    if (value[0]<2)
    {
        blegtr.scale[1]=2;
        blegtr.translation[0]=1;
    }
    else
    {
        blegtr.scale[1]=value[0];
        blegtr.translation[0]=value[0]/2;
    }
}
}"

# script for top leg length change (visual)
DEF LEG_LEN_CHANGE2 Script {
    directOutput TRUE
    eventIn SFVec3f trin
    field SFNode t11a USE T_LEG1A
    field SFNode t12a USE T_LEG2A
    field SFNode t13a USE T_LEG3A
    field SFNode t14a USE T_LEG4A
    field SFNode t15a USE T_LEG5A
    field SFNode t16a USE T_LEG6A
    url "javascript:
        function trin(value)
        {
            llen1=t11a.translation[0]+t11a.scale[1]/2;
            llen2=t12a.translation[0]+t12a.scale[1]/2;
            llen3=t13a.translation[0]+t13a.scale[1]/2;
            llen4=t14a.translation[0]+t14a.scale[1]/2;
            llen5=t15a.translation[0]+t15a.scale[1]/2;
            llen6=t16a.translation[0]+t16a.scale[1]/2;

            if (value[0]<1)
            {
                t11a.scale[1]=1;
                t11a.translation[0]=llen1-0.5;
                t12a.scale[1]=1;
                t12a.translation[0]=llen2-0.5;
                t13a.scale[1]=1;
                t13a.translation[0]=llen3-0.5;
                t14a.scale[1]=1;
                t14a.translation[0]=llen4-0.5;
                t15a.scale[1]=1;
                t15a.translation[0]=llen5-0.5;
                t16a.scale[1]=1;
                t16a.translation[0]=llen6-0.5;
            }
            else
            {
                t11a.scale[1]=value[0];
                t11a.translation[0]=llen1-value[0]/2;
                t12a.scale[1]=value[0];
                t12a.translation[0]=llen2-value[0]/2;
                t13a.scale[1]=value[0];
                t13a.translation[0]=llen3-value[0]/2;
                t14a.scale[1]=value[0];
                t14a.translation[0]=llen4-value[0]/2;
                t15a.scale[1]=value[0];
                t15a.translation[0]=llen5-value[0]/2;
                t16a.scale[1]=value[0];
                t16a.translation[0]=llen6-value[0]/2;
            }
        }
    }
}"

# main connection gate to the Java Applet
DEF EAL_CONTROL Script {
    directOutput TRUE

```

```

eventIn SFVec3f translation
eventIn SFRotation rotation
eventIn SFVec3f motion_reset
eventIn SFFloat triangular_top_set
eventIn SFVec3f hexagonal_top_set
eventIn SFBool custom_top_set
eventIn SFFloat triangular_base_set
eventIn SFVec3f hexagonal_base_set
eventIn SFBool custom_base_set
eventIn SFBool connector_length
field SFNode top_platform USE TOP_PLATFORM
field SFNode t11 USE T_LEG1
field SFNode t12 USE T_LEG2
field SFNode t13 USE T_LEG3
field SFNode t14 USE T_LEG4
field SFNode t15 USE T_LEG5
field SFNode t16 USE T_LEG6
field SFNode b11 USE B_LEG1
field SFNode b12 USE B_LEG2
field SFNode b13 USE B_LEG3
field SFNode b14 USE B_LEG4
field SFNode b15 USE B_LEG5
field SFNode b16 USE B_LEG6
field SFNode pst1 USE TS_T1
field SFNode pst2 USE TS_T2
field SFNode pst3 USE TS_T3
field SFNode pst4 USE TS_T4
field SFNode pst5 USE TS_T5
field SFNode pst6 USE TS_T6
field SFNode psb1 USE TS_B1
field SFNode psb2 USE TS_B2
field SFNode psb3 USE TS_B3
field SFNode psb4 USE TS_B4
field SFNode psb5 USE TS_B5
field SFNode psb6 USE TS_B6
field SFNode tlegsr USE TLEG_SENSOR
field SFNode blegsr USE BLEG_SENSOR
field SFNode tbc USE TOP_BALL_COLOR
field SFNode bbc USE BASE_BALL_COLOR

url *javascript:
function translation(value)
{
    top_platform.translation=value;
}
function rotation(value)
{
    top_platform.rotation=top_platform.rotation.multiply(value);
}
function motion_reset(value)
{
    top_platform.translation[0]=value[0];
    top_platform.translation[1]=value[1];
    top_platform.translation[2]=value[2];
    //top_platform.rotation[0]=0.0;
    //top_platform.rotation[1]=0.0;
    //top_platform.rotation[2]=1.0;
    //top_platform.rotation[3]=0.0;
}
function triangular_top_set(value)
{
    t11.translation[0]=pst1.offset[0]=0.5*value;
    t11.translation[1]=pst1.offset[1]=0.2887*value;
    t11.translation[2]=pst1.offset[2]=0.0;

    t12.translation[0]=pst2.offset[0]=0.5*value;
    t12.translation[1]=pst2.offset[1]=0.2887*value;
    t12.translation[2]=pst2.offset[2]=0.0;

    t13.translation[0]=pst3.offset[0]=-0.5*value;
    t13.translation[1]=pst3.offset[1]=0.2887*value;

```

```

t13.translation[2]=pst3.offset[2]=0.0;

t14.translation[0]=pst4.offset[0]=-0.5*value;
t14.translation[1]=pst4.offset[1]=0.2887*value;
t14.translation[2]=pst4.offset[2]=0.0;

t15.translation[0]=pst5.offset[0]=0.0;
t15.translation[1]=pst5.offset[1]=-0.5774*value;
t15.translation[2]=pst5.offset[2]=0.0;

t16.translation[0]=pst6.offset[0]=0.0;
t16.translation[1]=pst6.offset[1]=-0.5774*value;
t16.translation[2]=pst6.offset[2]=0.0;

tbc.diffuseColor[0]=0.7;
tbc.diffuseColor[1]=0.7;
tbc.diffuseColor[2]=0.7;

pst1.enabled = FALSE;
pst2.enabled = FALSE;
pst3.enabled = FALSE;
pst4.enabled = FALSE;
pst5.enabled = FALSE;
pst6.enabled = FALSE;
}

function triangular_base_set(value)
{
    bl1.translation[0]=psb1.offset[0]=0.5*value;
    bl1.translation[1]=psb1.offset[1]=-0.2887*value;
    bl1.translation[2]=psb1.offset[2]=0.0;

    bl2.translation[0]=psb2.offset[0]=0.0;
    bl2.translation[1]=psb2.offset[1]=0.5774*value;
    bl2.translation[2]=psb2.offset[2]=0.0;

    bl3.translation[0]=psb3.offset[0]=0.0;
    bl3.translation[1]=psb3.offset[1]=0.5774*value;
    bl3.translation[2]=psb3.offset[2]=0.0;

    bl4.translation[0]=psb4.offset[0]=-0.5*value;
    bl4.translation[1]=psb4.offset[1]=-0.2887*value;
    bl4.translation[2]=psb4.offset[2]=0.0;

    bl5.translation[0]=psb5.offset[0]=-0.5*value;
    bl5.translation[1]=psb5.offset[1]=-0.2887*value;
    bl5.translation[2]=psb5.offset[2]=0.0;

    bl6.translation[0]=psb6.offset[0]=0.5*value;
    bl6.translation[1]=psb6.offset[1]=-0.2887*value;
    bl6.translation[2]=psb6.offset[2]=0.0;

    bbc.diffuseColor[0]=0.6;
    bbc.diffuseColor[1]=0.7;
    bbc.diffuseColor[2]=0.6;

    psb1.enabled = FALSE;
    psb2.enabled = FALSE;
    psb3.enabled = FALSE;
    psb4.enabled = FALSE;
    psb5.enabled = FALSE;
    psb6.enabled = FALSE;
}

function hexagonal_top_set(value)
{
    a=value[0];
    ap=value[1]/100;
    if (value[2]==0)
    {
        t11.translation[0]=pst1.offset[0]=0.5*(1-ap)*a;

```

```

t11.translation[1]=pst1.offset[1]=(0.2887-0.8660*ap)*a;

t12.translation[0]=pst2.offset[0]=(0.5-ap)*a;
t12.translation[1]=pst2.offset[1]=0.2887*a;

t13.translation[0]=pst3.offset[0]=(-0.5+ap)*a;
t13.translation[1]=pst3.offset[1]=0.2887*a;

t14.translation[0]=pst4.offset[0]=-0.5*(1-ap)*a;
t14.translation[1]=pst4.offset[1]=(0.2887-0.8660*ap)*a;

t15.translation[0]=pst5.offset[0]=-0.5*a*ap;
t15.translation[1]=pst5.offset[1]=(-0.5774+0.8660*ap)*a;

t16.translation[0]=pst6.offset[0]=0.5*a*ap;
t16.translation[1]=pst6.offset[1]=(-0.5774+0.8660*ap)*a;
}

if (value[2]==1)
{
    t11.translation[0]=pst1.offset[0]=0.5*(1-ap)*a;
    t11.translation[1]=pst1.offset[1]=(0.2887-0.8660*ap)*a;

    t12.translation[0]=pst2.offset[0]=0.5*a;
    t12.translation[1]=pst2.offset[1]=0.2887*a;

    t13.translation[0]=pst3.offset[0]=(-0.5+ap)*a;
    t13.translation[1]=pst3.offset[1]=0.2887*a;

    t14.translation[0]=pst4.offset[0]=-0.5*a;
    t14.translation[1]=pst4.offset[1]=0.2887*a;

    t15.translation[0]=pst5.offset[0]=-0.5*a*ap;
    t15.translation[1]=pst5.offset[1]=(-0.5774+0.8660*ap)*a;

    t16.translation[0]=pst6.offset[0]=0.0;
    t16.translation[1]=pst6.offset[1]=-0.5774*a;
}

if (value[2]==2)
{
    t11.translation[0]=pst1.offset[0]=0.5*a;
    t11.translation[1]=pst1.offset[1]=0.2887*a;

    t12.translation[0]=pst2.offset[0]=0.5*a*(1-ap);
    t12.translation[1]=pst2.offset[1]=0.2887*a*(1-ap);

    t13.translation[0]=pst3.offset[0]=-0.5*a;
    t13.translation[1]=pst3.offset[1]=0.2887*a;

    t14.translation[0]=pst4.offset[0]=-0.5*a*(1-ap);
    t14.translation[1]=pst4.offset[1]=0.2887*a*(1-ap);

    t15.translation[0]=pst5.offset[0]=0.0;
    t15.translation[1]=pst5.offset[1]=-0.5774*a;

    t16.translation[0]=pst6.offset[0]=0.0;
    t16.translation[1]=pst6.offset[1]=-0.5774*a*(1-ap);
}

t11.translation[2]=pst1.offset[2]=0.0;
t12.translation[2]=pst2.offset[2]=0.0;
t13.translation[2]=pst3.offset[2]=0.0;
t14.translation[2]=pst4.offset[2]=0.0;
t15.translation[2]=pst5.offset[2]=0.0;
t16.translation[2]=pst6.offset[2]=0.0;

tbc.diffuseColor[0]=0.7;
tbc.diffuseColor[1]=0.7;
tbc.diffuseColor[2]=0.7;

```

```

pst1.enabled = FALSE;
pst2.enabled = FALSE;
pst3.enabled = FALSE;
pst4.enabled = FALSE;
pst5.enabled = FALSE;
pst6.enabled = FALSE;
)

function hexagonal_base_set(value)
{
  b=value[0];
  bp=value[1]/100;
  if (value[2]==0)
  {
    bl1.translation[0]=psb1.offset[0]=0.5*(1-bp)*b;
    bl1.translation[1]=psb1.offset[1]=(-0.2887+0.8660*bp)*b;

    bl2.translation[0]=psb2.offset[0]=0.5*b*bp;
    bl2.translation[1]=psb2.offset[1]=(0.5774-0.8660*bp)*b;

    bl3.translation[0]=psb3.offset[0]=-0.5*b*bp;
    bl3.translation[1]=psb3.offset[1]=(0.5774-0.8660*bp)*b;

    bl4.translation[0]=psb4.offset[0]=-0.5*(1-bp)*b;
    bl4.translation[1]=psb4.offset[1]=(-0.2887+0.8660*bp)*b;

    bl5.translation[0]=psb5.offset[0]=(-0.5+bp)*b;
    bl5.translation[1]=psb5.offset[1]=-0.2887*b;

    bl6.translation[0]=psb6.offset[0]=(0.5-bp)*b;
    bl6.translation[1]=psb6.offset[1]=-0.2887*b;
  }

  if (value[2]==1)
  {
    bl1.translation[0]=psb1.offset[0]=0.5*b;
    bl1.translation[1]=psb1.offset[1]=-0.2887*b;

    bl2.translation[0]=psb2.offset[0]=0.5*b*bp;
    bl2.translation[1]=psb2.offset[1]=(0.5774-0.8660*bp)*b;

    bl3.translation[0]=psb3.offset[0]=0.0;
    bl3.translation[1]=psb3.offset[1]=0.5774*b;

    bl4.translation[0]=psb4.offset[0]=-0.5*(1-bp)*b;
    bl4.translation[1]=psb4.offset[1]=(-0.2887+0.8660*bp)*b;

    bl5.translation[0]=psb5.offset[0]=-0.5*b;
    bl5.translation[1]=psb5.offset[1]=-0.2887*b;

    bl6.translation[0]=psb6.offset[0]=(0.5-bp)*b;
    bl6.translation[1]=psb6.offset[1]=-0.2887*b;
  }

  if (value[2]==2)
  {
    bl1.translation[0]=psb1.offset[0]=0.5*b;
    bl1.translation[1]=psb1.offset[1]=-0.2887*b;

    bl2.translation[0]=psb2.offset[0]=0.0;
    bl2.translation[1]=psb2.offset[1]=0.5774*b*(1-bp);

    bl3.translation[0]=psb3.offset[0]=0.0;
    bl3.translation[1]=psb3.offset[1]=0.5774*b;

    bl4.translation[0]=psb4.offset[0]=-0.5*b*(1-bp);
    bl4.translation[1]=psb4.offset[1]=-0.2887*b*(1-bp);

    bl5.translation[0]=psb5.offset[0]=-0.5*b;
    bl5.translation[1]=psb5.offset[1]=-0.2887*b;
  }
}

```

```

    bl6.translation[0]=psb6.offset[0]=0.5*b*(1-bp);
    bl6.translation[1]=psb6.offset[1]=-0.2887*b*(1-bp);
}

bbc.diffuseColor[0]=0.6;
bbc.diffuseColor[1]=0.7;
bbc.diffuseColor[2]=0.6;

psb1.enabled = FALSE;
psb2.enabled = FALSE;
psb3.enabled = FALSE;
psb4.enabled = FALSE;
psb5.enabled = FALSE;
psb6.enabled = FALSE;
}

function custom_top_set(value)
{
    tbc.diffuseColor[0]=1.0;
    tbc.diffuseColor[1]=1.0;
    tbc.diffuseColor[2]=0.0;

    pst1.enabled = TRUE;
    pst2.enabled = TRUE;
    pst3.enabled = TRUE;
    pst4.enabled = TRUE;
    pst5.enabled = TRUE;
    pst6.enabled = TRUE;
}

function custom_base_set(value)
{
    bbc.diffuseColor[0]=1.0;
    bbc.diffuseColor[1]=1.0;
    bbc.diffuseColor[2]=0.0;

    psb1.enabled = TRUE;
    psb2.enabled = TRUE;
    psb3.enabled = TRUE;
    psb4.enabled = TRUE;
    psb5.enabled = TRUE;
    psb6.enabled = TRUE;
}

function connector_length(value)
{
    tlegsr.enabled = value;
    blegsr.enabled = value;
}

}

# system event routing mechanism
ROUTE TOP_PLATFORM.translation TO LEGS.xyz
ROUTE TOP_PLATFORM.rotation TO LEGS.rxyz

ROUTE B_LEG1.translation_changed TO LEGS.bxyz
ROUTE B_LEG2.translation_changed TO LEGS.bxyz
ROUTE B_LEG3.translation_changed TO LEGS.bxyz
ROUTE B_LEG4.translation_changed TO LEGS.bxyz
ROUTE B_LEG5.translation_changed TO LEGS.bxyz
ROUTE B_LEG6.translation_changed TO LEGS.bxyz

ROUTE T_LEG1.translation_changed TO LEGS.txyz
ROUTE T_LEG2.translation_changed TO LEGS.txyz
ROUTE T_LEG3.translation_changed TO LEGS.txyz
ROUTE T_LEG4.translation_changed TO LEGS.txyz
ROUTE T_LEG5.translation_changed TO LEGS.txyz
ROUTE T_LEG6.translation_changed TO LEGS.txyz

ROUTE TS_B1.translation_changed TO B_LEG1.translation

```

```

ROUTE TS_B2.translation_changed TO B_LEG2.translation
ROUTE TS_B3.translation_changed TO B_LEG3.translation
ROUTE TS_B4.translation_changed TO B_LEG4.translation
ROUTE TS_B5.translation_changed TO B_LEG5.translation
ROUTE TS_B6.translation_changed TO B_LEG6.translation

ROUTE TS_T1.translation_changed TO T_LEG1.translation
ROUTE TS_T2.translation_changed TO T_LEG2.translation
ROUTE TS_T3.translation_changed TO T_LEG3.translation
ROUTE TS_T4.translation_changed TO T_LEG4.translation
ROUTE TS_T5.translation_changed TO T_LEG5.translation
ROUTE TS_T6.translation_changed TO T_LEG6.translation

ROUTE BLEG_SENSOR.translation_changed TO LEG_LEN_CHANGE.trin
ROUTE TLEG_SENSOR.translation_changed TO LEG_LEN_CHANGE2.trin

```

## 7. RUN.HTML

```

<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">
<title> Spatial Platform </title>
</head>

<body>

<h4 align="center"><font color="#008080" size="6" face="Tahoma"><b>Platform
Analysis</b></font></h4>
<div align="center"><center>

<table border="0">
  <tr>
    <td><embed src="Platform\Platform.wrl" align="baseline" border="0"
      width="500" height="520"></td>
    <td><applet code="Platform.Platform.class" align="baseline"
      width="250" height="520" mayscript></applet></td>
  </tr>
</table>
</center>
</body>
</html>

```

## 8. QIDB.ASP

```

<%
Response.Expires=-1

'.....
' connecting to the database
'.....

Dim Connect
Set Connect = Server.CreateObject("ADODB.Connection")
Connect.Open "qi"

'.....
' read the platform configuration from URL
'.....

```

```

topconfig = Request("top")
baseconfig = Request("base")

'*****
' replace the 'x' to spaces
'*****

topconfig=Replace(topconfig,"x"," ")
baseconfig=Replace(baseconfig,"x"," ")

'*****
' read a set of data from database
'*****

Dim qidb
Set qidb = Server.CreateObject("ADODB.RecordSet")
SQL = "SELECT * FROM QualityIndex " & _
      "Where TopConfig = '" & topconfig & "' AND" & _
      "      BaseConfig = '" & baseconfig & "' AND status = 'DONE'"
qidb.Open SQL, Connect

If Not qidb.EOF Then
    '** if there is a matching record set, return the result
    Response.Write("FOUND" & vbNewLine)
    Response.Write(cdbl(qidb("maxdet")) & vbNewLine)
    Response.Write(cdbl(qidb("maxxx")) & vbNewLine)
    Response.Write(cdbl(qidb("maxy")) & vbNewLine)
    Response.Write(cdbl(qidb("maxz")) & vbNewLine)
    Response.Write(cdbl(qidb("maxrx")) & vbNewLine)
    Response.Write(cdbl(qidb("maxry")) & vbNewLine)
    Response.Write(cdbl(qidb("maxrz")) & vbNewLine)
Else
    '** checking any other search program is running
    Dim process
    Set process = Server.CreateObject("ADODB.RecordSet")
    SQL="SELECT * FROM QualityIndex WHERE status = 'PROCESSING'"
    process.Open SQL, Connect

    if process.EOF Then
        '** no other process is running so insert the config data to the database
        '** and call the search program.
        SQL="INSERT INTO qualityIndex VALUES ('" & topconfig & "','" & baseconfig & _
          "' & ",-1,0,0,0,0,0,0,'PROCESSING')'"
        Connect.Execute(SQL)

        Set WshShell = CreateObject( "WScript.Shell" )
        WshShell.Run "D:\JHL\QIDB\RELEASE\MAXDET.EXE"
        Response.Write("PROCESSING" & vbNewLine)

    Else
        '** another process is running so wait till it finishes
        Response.Write("WAIT" & vbNewLine)

    End If

    process.Close

End If

qidb.Close

Connect.Close

%>

```



## 9. MaxDet.cpp

```

#import "C:\Program Files\Common Files\System\ADO\msado15.dll" \
    no_namespace rename("EOF", "EndOfFile")

#include <stdio.h>
#include <math.h>
#include <ole2.h>
#include <malloc.h>
#include <conio.h>
#include "icrsint.h"

#define PI 3.141592

// Function Declaration.
inline void TESTHR(HRESULT x) (if FAILED(x) _com_issue_error(x));
void PrintProviderError(_ConnectionPtr pConnection);
void PrintComError(_com_error &e);
int searching_maxdet(char* topconfig, char* baseconfig, double result[7]);
void update_database(void);
void get_jacobian(double p[12][3], double jmat[6][6]);
double get_determinant(double J[6][6]);
double search(double p[12][3], double maxcoord[6]);
void apply_xf(double p[12][3], double x, double y, double z,
              double rx, double ry, double rz, double c[12][3]);
void vec_mult(double p[4], double m[4][4]);
void mat_mult(double m1[4][4], double m2[4][4], double r[4][4]);
void display_matrix(double m[4][4]);

//
//
//      Main Function
//
//
//
void main(void)
{
    HRESULT hr = S_OK;

    if(FAILED(::CoInitialize(NULL)))
        return;

    update_database();

    ::CoUninitialize();
}

//
//
//      Read from the database and update the data
//
//
//
void update_database()
{
    double result[7];
    char buffer[11];
    int ok;

    // Define ADO object pointers.
    // Initialize pointers on define.
    // These are in the ADOB:: namespace.
    _RecordsetPtr pRstQI = NULL;

    // Define string variables.
    _bstr_t strConn("Data Source=q1;User Id=;Password=;");

    try
    {

```

```

// Open recordset with names from QI table.
TESTHR(pRstQI.CreateInstance(__uuidof(Recordset)));
pRstQI->CursorType = adOpenKeyset;
pRstQI->LockType = adLockOptimistic;
pRstQI->Open("SELECT * FROM QualityIndex "
    "WHERE status = 'PROCESSING'",strCnn,adOpenKeyset,adLockOptimistic,
    adCmdText);

// Store original data.
_bstr_t strTopConfig = pRstQI->Fields->GetItem("topconfig")->Value;
_bstr_t strBaseConfig = pRstQI->Fields->GetItem("baseconfig")->Value;

_bstr_t strMaxDet = pRstQI->Fields->GetItem("maxdet")->Value;
_bstr_t strMaxX = pRstQI->Fields->GetItem("maxx")->Value;
_bstr_t strMaxY = pRstQI->Fields->GetItem("maxy")->Value;
_bstr_t strMaxZ = pRstQI->Fields->GetItem("maxz")->Value;
_bstr_t strMaxRX = pRstQI->Fields->GetItem("maxrx")->Value;
_bstr_t strMaxRY = pRstQI->Fields->GetItem("maxry")->Value;
_bstr_t strMaxRZ = pRstQI->Fields->GetItem("maxrz")->Value;
_bstr_t strStatus = pRstQI->Fields->GetItem("status")->Value;

ok = searching_maxdet((char*)strTopConfig, (char*)strBaseConfig, result);

//Change data in edit buffer.
_gcvt(result[0],10,buffer);
pRstQI->Fields->GetItem("maxdet")->Value = (_bstr_t)(buffer);
_gcvt(result[1],10,buffer);
pRstQI->Fields->GetItem("maxx")->Value = (_bstr_t)(buffer);
_gcvt(result[2],10,buffer);
pRstQI->Fields->GetItem("maxy")->Value = (_bstr_t)(buffer);
_gcvt(result[3],10,buffer);
pRstQI->Fields->GetItem("maxz")->Value = (_bstr_t)(buffer);
_gcvt(result[4],10,buffer);
pRstQI->Fields->GetItem("maxrx")->Value = (_bstr_t)(buffer);
_gcvt(result[5],10,buffer);
pRstQI->Fields->GetItem("maxry")->Value = (_bstr_t)(buffer);
_gcvt(result[6],10,buffer);
pRstQI->Fields->GetItem("maxrz")->Value = (_bstr_t)(buffer);
pRstQI->Fields->GetItem("status")->Value = (_bstr_t)("DONE");

if (ok==0) pRstQI->Update();
else pRstQI->CancelUpdate();

pRstQI->Close();
}

catch(_com_error &e)
{
    // Notify the user of errors if any.
    // Pass a connection pointer accessed from the Recordset.
    _variant_t vtConnect = pRstQI->GetActiveConnection();

    // GetActiveConnection returns connect string if connection
    // is not open, else returns Connection object.
    switch(vtConnect.vt)
    {
        case VT_BSTR:
            PrintComError(e);
            break;
        case VT_DISPATCH:
            PrintProviderError(vtConnect);
            break;
        default:
            printf("Errors occurred.");
            break;
    }
}

}

//
//

```

```

//      PrintProviderError Function                                     //
//      ///////////////////////////////////////////////////////////////////
void PrintProviderError(_ConnectionPtr pConnection)
{
    // Print Provider Errors from Connection object.
    // pErr is a record object in the Connection's Error collection.
    ErrorPtr    pErr = NULL;

    if( (pConnection->Errors->Count) > 0)
    {
        long nCount = pConnection->Errors->Count;
        // Collection ranges from 0 to nCount -1.
        for(long i = 0; i < nCount; i++)
        {
            pErr = pConnection->Errors->GetItem(i);
            printf("Error number: %x\t%s\n", pErr->Number,
                (LPCSTR) pErr->Description);
        }
    }
}

//      ///////////////////////////////////////////////////////////////////
//      ///////////////////////////////////////////////////////////////////
//      PrintComError Function                                         //
//      ///////////////////////////////////////////////////////////////////
void PrintComError(_com_error &e)
{
    _bstr_t bstrSource(e.Source());
    _bstr_t bstrDescription(e.Description());

    // Print Com errors.
    printf("Error\n");
    printf("\tCode = %08lx\n", e.Error());
    printf("\tCode meaning = %s\n", e.ErrorMessage());
    printf("\tSource = %s\n", (LPCSTR) bstrSource);
    printf("\tDescription = %s\n", (LPCSTR) bstrDescription);
}

//      ///////////////////////////////////////////////////////////////////
//      ///////////////////////////////////////////////////////////////////
//      DetJm search Function                                          //
//      ///////////////////////////////////////////////////////////////////
int searching_maxdet(char* topconfig, char* baseconfig, double result[7])
{
    double p[12][3], maxcoord[6];
    int i;

    sscanf(topconfig, "%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf",
        &p[0][0], &p[0][1], &p[1][0], &p[1][1], &p[2][0], &p[2][1],
        &p[3][0], &p[3][1], &p[4][0], &p[4][1], &p[5][0], &p[5][1]);

    sscanf(baseconfig, "%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf",
        &p[6][0], &p[6][1], &p[7][0], &p[7][1], &p[8][0], &p[8][1],
        &p[9][0], &p[9][1], &p[10][0], &p[10][1], &p[11][0], &p[11][1]);

    for (i=0; i<6; i++)
    {
        p[i][0]=p[i][0]/10000.0;
        p[i][1]=p[i][1]/10000.0;
        p[i][2]=0.0;
        p[i+6][0]=p[i+6][0]/10000.0;
        p[i+6][1]=p[i+6][1]/10000.0;
        p[i+6][2]=0.0;
    }

    result[0]=ceil(search(p,maxcoord)*100)/100;
}

```

```

for(i=1; i<=3; i++) result[i]=ceil(maxcoord[i-1]*1000)/1000;
for(i=4; i<=6; i++) result[i]=ceil(maxcoord[i-1]*10)/10;

return 0;
}

//////////////////////////////////////
//                               //
//      DetJm search sub function      //
//                               //
//////////////////////////////////////
double search(double p[12][3], double maxcoord[6])
{
    double jmat[6][6], detj, coords[12][3];
    double x,y,z, rx, ry, rz;
    double minv[6], maxv[6], tstep, astep;
    double maxdet;
    int i, ld;

    maxdet=0.0;

    minv[0] = minv[1] = -2;   minv[2] = 0.0;
    maxv[0] = maxv[1] = 2;    maxv[2] = 4;
    tstep = 1;

    minv[3] = minv[4] = minv[5] = -180;
    maxv[3] = maxv[4] = maxv[5] = 180;
    astep = 90;

    ld=0;
    while(tstep>0.0001 && astep>0.01)
    {
        ld++;
        printf("Loop in %d   tstep : %f   astep : %f\n\n", ld, tstep, astep);
        rx=0;ry=0;
        for (rx=minv[3]; rx<maxv[3]; rx+=astep) {
            for (ry=minv[4]; ry<maxv[4]; ry+=astep) {
                for ( x=minv[0]; x<=maxv[0]; x+=tstep) {
                    for ( y=minv[1]; y<=maxv[1]; y+=tstep) {
                        for ( z=minv[2]; z<=maxv[2]; z+=tstep) {
                            for (rz=minv[5]; rz<maxv[5]; rz+=astep) {

                                apply_xf(p,x,y,z, rx, ry, rz, coords);
                                get_jacobian(coords, jmat);
                                detj = fabs(get_determinant(jmat));
                                if (detj>maxdet)
                                {
                                    maxdet=detj;
                                    maxcoord[0]=x; maxcoord[1]=y; maxcoord[2]=z;
                                    maxcoord[3]=rx; maxcoord[4]=ry; maxcoord[5]=rz;
                                }
                            }
                        }
                    }
                }
            }
        }

        for(i=0; i<3; i++)
        {
            minv[i]=maxcoord[i]-tstep;
            minv[i+3]=maxcoord[i+3]-astep;
            maxv[i]=maxcoord[i]+tstep;
            maxv[i+3]=maxcoord[i+3]+astep;
        }

        printf("max = %f\n", maxdet);
        printf("Xyz : %f %f %f\n", maxcoord[0], maxcoord[1], maxcoord[2]);
        printf("rxyz : %f %f %f\n", maxcoord[3], maxcoord[4], maxcoord[5]);

        tstep = tstep / 4;
        astep = astep / 4;

        if (minv[2]<0) minv[2]=0;
    }
}

```

```

    return maxdet;
}

////////////////////////////////////
//
//   Apply transformation matrix to the top platform   //
//
////////////////////////////////////
void apply_xf(double p[12][3], double x, double y, double z,
              double rx, double ry, double rz, double c[12][3])
{
    rx=rx*PI/180;
    ry=ry*PI/180;
    rz=rz*PI/180;

    double ptr[4],mat[4][4],tmpa[4][4],tmpb[4][4];
    double mt[4][4]={1,0,0,x},
               {0,1,0,y},
               {0,0,1,z},
               {0,0,0,1};
    double mrx[4][4]={1,      0,      0,0},
               {0, cos(rx), -sin(rx),0},
               {0, sin(rx),  cos(rx),0},
               {0,      0,      0,1};
    double mry[4][4]={ cos(ry), 0, sin(ry),0},
               {0, 1,      0,0},
               {-sin(ry), 0, cos(ry),0},
               {0, 0,      0,1};
    double mrz[4][4]={cos(rz),-sin(rz),0,0},
               {sin(rz), cos(rz),0,0},
               {0,0,1,0},
               {0,0,0,1};

    int i,j;

    mat_mult(mt,mrx,tmpa);
    mat_mult(tmpa,mry,tmpb);
    mat_mult(tmpb,mrz,mat);

    ptr[3]=1.0;
    for(i=0; i<6; i++)
    {
        for(j=0;j<3;j++) ptr[j]=p[i][j];
        vec_mult(ptr,mat);
        for(j=0;j<3;j++) c[i][j]=ptr[j];
    }

    for(i=6; i<12; i++)
        for(j=0; j<3; j++) c[i][j]=p[i][j];
}

////////////////////////////////////
//
//   Matrix-Vector Multiplication (4x4) (4x1)   //
//
////////////////////////////////////
void vec_mult(double p[4], double m[4][4])
{
    int i,j;
    double r[4],value;

    for(i=0; i<4; i++)
    {
        value=0.0;
        for(j=0; j<4; j++)
            value+=m[i][j]*p[j];
        r[i]=value;
    }
    for(i=0; i<4; i++) p[i]=r[i];
}

```

```

//////////////////////////////////////////////////////////////////
//
//      Matrix-Matrix Multiplication (4x4)
//
//////////////////////////////////////////////////////////////////
void mat_mult(double m1[4][4], double m2[4][4], double r[4][4])
{
    int i,j,k;
    double value;

    for(i=0; i<4; i++)
        for(j=0; j<4; j++)
        {
            value=0.0;
            for(k=0; k<4; k++) value+=m1[i][k]*m2[k][j];
            r[i][j]=value;
        }
}

//////////////////////////////////////////////////////////////////
//
//      Form a Jacobian matrix
//
//////////////////////////////////////////////////////////////////
void get_jacobian(double p[12][3], double jmat[6][6])
{
    double l;

    for (int i=0; i<6; i++)
    {
        l = sqrt((p[i][0]-p[i+6][0])*(p[i][0]-p[i+6][0])
            +(p[i][1]-p[i+6][1])*(p[i][1]-p[i+6][1])+(p[i][2]-p[i+6][2])
            *(p[i][2]-p[i+6][2]));
        jmat[0][i] = (p[i][0]-p[i+6][0])/l;
        jmat[1][i] = (p[i][1]-p[i+6][1])/l;
        jmat[2][i] = (p[i][2]-p[i+6][2])/l;
        jmat[3][i] = (p[i][1]*p[i+6][2]-p[i][2]*p[i+6][1])/l;
        jmat[4][i] = (p[i][2]*p[i+6][0]-p[i][0]*p[i+6][2])/l;
        jmat[5][i] = (p[i][0]*p[i+6][1]-p[i][1]*p[i+6][0])/l;
    }
}

//////////////////////////////////////////////////////////////////
//
//      Calculate determinant of J
//
//////////////////////////////////////////////////////////////////
double get_determinant(double J[6][6])
{
    double ans=0;
    double s[6], D[5][6];
    int i,j,k,l,m,n;

    s[0]=0;
    for(i=0;i<6;i++)
    {
        s[1]=0;
        D[0][i]=0;
        for(j=0;j<6;j++)
        {
            if(j==i) continue;
            s[2]=0;
            D[1][j]=0;
            for(k=0;k<6;k++)
            {
                if(k==i||k==j) continue;
                s[3]=0;

```

```

D[2][k]=0;
for(1=0;1<6;1++)
{
    if(1==i||1==j||1==k) continue;
    s[4]=0;
    D[3][1]=0;
    for(m=0;m<6;m++)
    {
        if(m==i||m==j||m==k||m==1) continue;
        s[5]=0;
        D[4][m]=0;
        for(n=0;n<6;n++)
        {
            if(n==i||n==j||n==k||n==1||n==m) continue;
            D[4][m]=pow(-1,s[5])*J[5][n];
            s[5]++;
        }
        D[3][1]+=pow(-1,s[4])*J[4][m]*D[4][m];
        s[4]++;
    }
    D[2][k]+=pow(-1,s[3])*J[3][1]*D[3][1];
    s[3]++;
}
D[1][j]+=pow(-1,s[2])*J[2][k]*D[2][k];
s[2]++;
}
D[0][i]+=pow(-1,s[1])*J[1][j]*D[1][j];
s[1]++;
}
ans+=pow(-1,s[0])*J[0][i]*D[0][i];
s[0]++;
}
return ans;
}

```

## LIST OF REFERENCES

- Angeles, J., and López-Cajún, C. 1992, "Kinematic isotrophy and the conditioning index of serial robotic manipulators", The International Journal of Robotics Research, vol 11, No. 6, p. 560-571.
- Beni, G., and Hackwood, S. 1985, "Recent Advances in Robotics," Chapter 5, Willy-Interscience Publication, New York.
- Carey, R., Bell, G., and Marrin, C. 1997, "Virtual Reality Modeling Language (VRML97)," ISO/IEC 14772-1:1997.
- Cox, D. J. 1981. "The dynamic modeling and command signal formulation for parallel multi-parameter robotic devices," Master thesis, University of Florida, Department of Mechanical Engineering.
- Daconta, M., Saganich. 1998, "Java 1.2 and Javascript for C and C++ Programmers," Wiley Computer Publishing, New York.
- Doubilet, P., Rota, G. C, and Stein, J. 1974, "On the foundations of combinatorial theory: IX, Combinatorial methods in invariant theory," Studies Appl. Math., 53, p. 185-216.
- Duffy, J. 1996. "Statics and Kinematics with Applications to Robotics," Cambridge University Press, New York.
- Fichter, E. F. 1986. "A Stewart platform-based manipulator: general theory and practical construction," International Journal of Robotic Research, Vol 5, No 2, pp. 157-182.
- Hunt, K. H. 1978. "Kinematic Geometry of Mechanisms," Oxford University Press, Oxford.
- Hunt, K. H., and McAreë, P. R. 1998. "The octahedral manipulator: geometry and mobility," International Journal of Robotic Research, Vol.17, No.8, pp.868-885.
- Keler, M. 1991, "Kriterien der führbarkeitseinschränkung von effektoren in robotern," Maschinenbautechnik, Berlin 40, p.155-158.

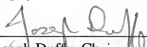


- Klein, C., and Blaho, B. 1987, "*Dexterity measures for the design and control of kinematically redundant manipulators*", The International Journal of Robotics Research, vol 6, No.2, p. 72-83.
- Lee, J. 1996. "*An Investigation of a quality index for the stability of in-parallel planar platform devices*," Master thesis, University of Florida, Department of Mechanical Engineering.
- Lee, J., Duffy, J., and Hunt, K. H. 1998. "*A practical quality index based on the octahedral manipulator*," International Journal of Robotic Research, Vol 17, No.10, pp. 1081-1090.
- Lee, J., Duffy, J., and Keler, M. 1996. "*The optimum quality index for the stability of in-parallel planar platform devices*," Proc. of ASME design engineering technical conferences, published on CDROM, 96-DETC/MECH-1135, Irvine, California.
- McAree, P. R., and Daniel, R. W. 1996. "*A fast, robust solution to the Stewart platform forward kinematics*," International Journal of Robotic Research, Vol 13, No.7, pp. 407-427.
- Martin, C., McCloskey, B., Sandvik, K., and Chin, D. 1997, "*Creating Interactive Java applications with 3D and VRML*," A Silicon Graphics, Inc, White Paper.
- McMillan III, T. 1990. "*Invariants of anti-symmetric tensors*," Ph.D. dissertation, University of Florida, Mathematics Department.
- Merlet, J., and Gosselin, C. 1991. "*Nouvelle architecture pour un manipulateur parallele a six degres de liberte*," Mech. Mach. Theory, Vol. 26, No 1, pp. 77-90.
- Stewart, D. 1965-6. "*A platform with six degree of freedom*," Proc. of I. Mech. E., Vol 180, part 1, pp. 371-386.
- Stoughton, R. S., and Arai, T. 1993. "*A modified Stewart platform manipulator with improved dexterity*," IEEE Trans. on Robotics and Automation, Vol 9, No 2, pp. 166-173.
- White, N. 1994, "*Grassmann-Cayley algebra and robotics*," Journal of Intelligent and Robotic Systems, 11, p. 91-107.
- White, N., and Whiteley, W. 1983. "*The algebraic geometry of stresses in frameworks*," SIAM J.Algebraic Discrete Methods Vol 4, No 4, pp 481-511.

## BIOGRAPHICAL SKETCH

Jaehoon Lee was born in Seoul, Korea, in 1967. He grew up in Seoul and graduated from the Sangmoon high school in 1986. After that he enrolled in the Hanyang University and received a Bachelor of Science degree in mechanical engineering in February, 1993. In April, 1989, he was enlisted in the army and he had served as a computer programmer for three years. In September, 1993, he enrolled in graduate school at the University of Florida. He has been working at the Center for Intelligent Machines and Robotics (CIMAR) as a graduate research assistant under the guidance of Dr. Joseph Duffy and Dr. Carl Crane since then. He received the master's degree in mechanical engineering at University of Florida in 1996.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
Joseph Duffy, Chair  
Graduate Research Professor of  
Mechanical Engineering

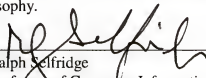
I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
Carl D. Crane  
Professor of Mechanical Engineering

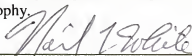
I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
John C. Ziegert  
Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
Ralph Selfridge  
Professor of Computer Information  
Science and Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

  
\_\_\_\_\_  
Neil White  
Professor of Mathematics

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 2000

A handwritten signature in dark ink, appearing to read 'M. J. Ohanian', is written over a horizontal line.

M. J. Ohanian  
Dean, College of Engineering

---

Winfred M. Phillips  
Dean, Graduate School